

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

BAKALÁŘSKÁ PRÁCE

Brno, 2018

Štefan Krajanec



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ

DEPARTMENT OF TELECOMMUNICATIONS

MOBILNÍ APLIKACE VYUŽÍVAJÍCÍ TECHNOLOGII GOOGLE BEACON

MOBILE APPLICATION USING GOOGLE BEACON TECHNOLOGY

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Štefan Krajanec

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Petr Dzurenda

BRNO 2018

Bakalářská práce

bakalářský studijní obor **Teleinformatika**

Ústav telekomunikací

Student: Štefan Krajanec

ID: 186120

Ročník: 3

Akademický rok: 2017/18

NÁZEV TÉMATU:

Mobilní aplikace využívající technologii Google beacon

POKYNY PRO VYPRACOVÁNÍ:

Seznamte se se současnými technologiemi pro lokalizaci a identifikaci osob. Důraz kladte na využití technologie Google beacon pro marketingové účely v rámci obchodních center. Navrhněte a implementujte ucelený systém pro analýzu chování uživatelů, na jejímž základě bude uživatel informován o různých slevových akcích či jiných událostech.

DOPORUČENÁ LITERATURA:

[1] MURPHY, Mark L. Android 2: Průvodce programováním mobilních aplikací. Vyd. 1. Brno: Computer Press, 2011, 375 s. ISBN 978-80-251-3194-7.

[2] Google Beacon Platform. Google developers [online]. [cit. 2017-09-11]. Dostupné z: <https://developers.google.com/beacons/>

Termín zadání: 5.2.2018

Termín odevzdání: 29.5.2018

Vedoucí práce: Ing. Petr Dzurenda

Konzultant:

prof. Ing. Jiří Mišurec, CSc.
předseda oborové rady

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

ABSTRAKT

Bakalárska práca sa zaoberá využitím beacon technológie v Android aplikáciách. Beacon technológia umožňuje určiť približnú vzdialenosť užívateľa od beaconu. Za účelom overenia hodnôt udaných výrobcom bolo uskutočnené meranie na zistenie závislosti prijatého výkonu na vzdialenosti. Merané boli tri značky beacnov Kontakt.io, PROXioT a IBKS, meranie bolo uskutočnené v otvorenom prostredí. Zistenia z merania boli implementované do aplikácie zastrešujúcej cieľný marketing v obchodných centrách. Aplikácia po priblížení k beacnu rozlišuje niekoľko typov správania beaconu. Jedná sa o správanie ako okamžitá notifikácia, zápis polohy do databázy, sledovanie a zápis približného času stráveného pri beacne do databázy a podmienenú odpoveď. Snahou bolo vyriešiť problematiku komunikácie s beacnom čo najuniverzálnejšie, aby správa a implementácia do ďalších projektov bola čo najjednoduchšia. Spomenutá aplikácia je výstupom tejto práce.

KĽÚČOVÉ SLOVÁ

Beacon, Eddystone, iBeacon, Bluetooth Low Energy, Android

ABSTRACT

This Bachelor thesis focuses on utilizing beacon technology in Android applications. Beacon technology allows for determining an approximate distance of the user from the beacon.. In order to verify the reference numbers given by the manufacturer we conducted an observation to measure the power of the received signal in reference to distance. 3 brands of beacons were tested, Kontakt.io, PROXioT and IBKS. The observation has been iterated in an open field setting. Findings from this observation was implemented into an application designed for shopping mall marketing. The application, after getting closer to the beacon distinguishes between different behaviour types; behaviour as an instant notification, as a position log into a database, as a time stamp log (time spent near the beacon) into a database and as a conditional response. The aim was to resolve the problematics of communication with the beacon as universally as possible so the implemtation into other projects would be as simple as possible. The aforementioned app is the output of this thesis.

KEYWORDS

Beacon, Eddystone, iBeacon, Bluetooth Low Energy, Android

KRAJANEC, Štefan. *Mobilní aplikace využívající technologii Google beacon*. Brno, 2018, 53 s. Bakalárska práca. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Vedúci práce: Ing. Petr Dzurenda,

VYHLÁSENIE

Vyhlasujem, že som svoju bakalársku prácu na tému „Mobilní aplikace využívající technologii Google beacon“ vypracoval(a) samostatne pod vedením vedúceho bakalárskej práce, využitím odbornej literatúry a ďalších informačných zdrojov, ktoré sú všetky citované v práci a uvedené v zozname literatúry na konci práce.

Ako autor(ka) uvedenej bakalárskej práce ďalej vyhlasujem, že v súvislosti s vytvorením tejto bakalárskej práce som neporušil(a) autorské práva tretích osôb, najmä som nezasiahol(-la) nedovoleným spôsobom do cudzích autorských práv osobnostných a/alebo majetkových a som si plne vedomý(-á) následkov porušenia ustanovenia § 11 a nasledujúcich autorského zákona Českej republiky č. 121/2000 Sb., o práve autorskom, o právach súvisiacich s právom autorským a o zmene niektorých zákonov (autorský zákon), v znení neskorších predpisov, vrátane možných trestnoprávných dôsledkov vyplývajúcich z ustanovenia časti druhej, hlavy VI. diel 4 Trestného zákoníka Českej republiky č. 40/2009 Sb.

Brno

.....

podpis autora(-ky)

POĎAKOVANIE

Rád by som poďkoval vedúcemu bakalárskej práce pánovi Ing. Petrovi Dzurendovi, za odborné vedenie, konzultácie, trpezlivosť a podnetné návrhy k práci. Ďalej by som rád poďakoval obchodným centrám OC Mlyny a OC Max v Nitre za umožnenie merania v ich priestoroch.

Brno

.....

podpis autora(-ky)

OBSAH

Úvod	10
1 Identifikačné technológie	11
1.1 Čiarový kód (Barcode)	11
1.2 QR code	11
1.3 RFID tagy	12
1.3.1 LF tagy	12
1.3.2 HF tagy	13
1.3.3 UHF tag	13
1.4 Beacon	14
1.4.1 Rôzne druhy beacnov	15
1.4.2 iBeacon	15
1.4.3 Eddystone	16
1.4.4 Testované zariadenia	18
1.4.5 Možnosti využitia technológie v reálnom svete	19
2 Technológie použité v aplikácii	22
2.1 Android studio	22
2.2 MySQL	23
2.3 PHP	23
2.4 JSON	23
2.5 Firebase	24
2.6 Realm	24
3 Návrh a implementácia aplikácií	25
3.1 Meranie intenzity vysielaného výkonu	25
3.1.1 Vyhodnotenie merania	27
3.2 Aplikácia zastrešujúca marketing v obchodoch	30
3.2.1 Testovací scenár	30
3.2.2 MySQL Databáza	32
3.2.3 Rozdelenie Android aplikácie na časti - moduly	35
3.2.4 Autentizácia	40
3.2.5 Spracovanie nájdeného beacnu	42
3.2.6 Prijatie a spracovanie notifikácie	46
3.3 Komunikácia v aplikácii	47
4 Záver	49

Literatúra	50
Zoznam symbolov, veličín a skratiek	51
Zoznam príloh	52
A Obsah přiloženého CD	53

ZOZNAM OBRÁZKOV

1.1	Príklad Barkódu	11
1.2	Príklad QR kódu	12
1.3	Ukážka NFC tagu	14
1.4	Rôzne modely Beaconov	15
1.5	Demonštrácia dátových blokov v protokole iBeacon [2]	16
1.6	Demonštrácia dátových blokov v protokole Eddystone [1]	17
1.7	Fotka testovaných zariadení od Kontakt.io (Eddystone iBeacon), Proxiot (Eddystone iBeacon), iBKS (iBeacon)	20
3.1	Vizuálna ukážka aplikácie na meranie intenzity signálu	27
3.2	Zobrazenie postupnosti komunikácie medzi beacnom, telefónom a da- tabázou	27
3.3	Diagram opisujúci komunikáciu beacon -> smartfón -> server pri meraní	28
3.4	Závislosť prijatého výkonu vzhľadom na vzdialenosť	28
3.5	Chybovosť prijatých správ z beacnu vzhľadom na vzdialenosť od beacnu	29
3.6	Názorné rozmiestnenie beaconov v jednoduchom obchode	31
3.7	Možnosti pohybu návštevníka	31
3.8	SQL príkaz na tvorbu tabuľky užívateľov	32
3.9	SQL príkaz na tvorbu tabuľky tagov	34
3.10	Znázornenie závislostí modulov v projekte	36
3.11	Registrácia nového užívateľa do aplikácie	37
3.12	Prihlásenie užívateľa do aplikácie	37
3.13	Postup komunikácie prihlásenia	37
3.14	Detail obdržanej notifikácie	38
3.15	Aktivita zobrazujúca všetky obdržané kupóny	39
3.16	Znázornenie komunikácie v sekvenčnom diagrame	39
3.17	PHP skript registrácia	42
3.18	Funkcia skriptu tvoriaceho notifikáciu	46
3.19	Komunikácia medzi beacnami BEACON2 a BEACON3 so serverom .	48
3.20	Komunikácia medzi beacnom BEACON1 a serverom	48

ZOZNAM TABULIEK

1.1	Možnosti upravenia výkonu Kontakt.io beaconu	19
1.2	Porovnanie beacnov rôznych výrobcov	19
3.1	Bližšie informácie k tabulke užívateľov	33
3.2	Bližšie informácie k tabulke tagov	34

ÚVOD

Technológia beacon nie je ešte moc známa, jedná sa o technológiu prvýkrát predstavenú v roku 2013. Táto technológia má mnoho spôsobov využitia, jedná sa o najmladší prírastok medzi identifikačnými technológiami. V prvej kapitole je možné sa dočítať o starších technológiách ktoré sa stále používajú na identifikáciu. Táto práca sa venuje vývoju marketingovej aplikácie spolupracujúcej s technológiou beacon. Najskôr je nutné zoznámiť sa s technológiou beaconov, keďže je nosným bodom tejto práce.

Technológia beacon je založená na periodickom vysielaní dát do okolia prostredníctvom Bluetooth Low Energy, ktorým disponuje už väčšina smartfónov. Smartfónu je umožnené prečítať informáciu ktorú beacon vysielá a na podnet dát z beaconu môže vykonať nejakú akciu. Beacons je možné nastaviť na viacero výkonnostných úrovní, ktoré udávajú vzdialenosť, do ktorej sú vyžarované informácie detegovateľné.

V súčasnosti každý človek vlastní najmenej jeden inteligentný telefón (smartfón), či sa jedná o Android alebo iOS. Obe tieto platformy implementujú okrem iných technológií aj technológiu Bluetooth Low Energy. Práve táto technológia je využívaná aj technológiou beacon. Pomocou využitia týchto dvoch technológií je možné vytvoriť logiku zastrešujúcu cieľový marketing v obchodných centrách. Cieľená reklama a analýza záujmov zákazníka je v aktuálnom období dosť rozvíjajúce sa odvetvie preto bola v tejto bakalárskej práci vytvorená aplikácia na Android implementujúca túto logiku.

Cieľom tejto bakalárskej práce je návrh a realizácia aplikácie určenej na smartfóny s OS Android, ktorá bude dostávať informácie z jednotlivých beaconov a ďalej ich spracovať. Na základe informácií z beaconov o pohybe užívateľa v predajni môže užívateľ dostať notifikáciou ponuku, ktorú môže okamžite využiť v predajni z ktorej odchádzal. Touto problematikou sa zaoberá posledná kapitola.

1 IDENTIFIKAČNÉ TECHNOLOGIE

Identifikačné technológie sú technológie na identifikáciu alebo overenie produktu. Používajú sa na zistenie informácie z nejakého media. Postupom času sa vytvorilo množstvo technológií, na ktorých boli rôzne média založené.

1.1 Čiarový kód (Barcode)

Barkód je väčšinou obdĺžnik tvorený sekvenciou čiar a medzier s definovanou šírkou, ktoré je možné prečítať čítačkou. Barkódy sa používajú na rýchlu a lacnú identifikáciu produktov. Je to jeden z najlacnejších spôsobov identifikácie, kód stačí vytlačiť na kus papiera, plastu, textilu alebo len zobraziť na displeji smartfónu. Na dekódovanie nič nehovoriacich čiar na užitočné identifikačné číslo, ktoré je nimi reprezentované, sa používa špeciálna čítačka, pomocou ktorej sa čiary naskenujú a porovnávajú s tabulkovými hodnotami daného kódovania. Ak je nájdená zhoda, výsledkom je pôvodná zakódovaná správa, ktorú je možné ďalej spracovať, najčastejšie na zistenie ceny tovaru, ktorý je priradený k danému kódu. Barkódom sa dajú zašifrovať len sekvencie číslíc. Na obr. 1.1 je vyobrazený príklad bar kódu.



Obr. 1.1: Príklad Barkódu

1.2 QR code

QR kód (Quick Response) je typ technológie, ktorá zakóduje informáciu do sekvencie bielych a tmavých štvorcov do jedného väčšieho štvorca. Ide o novšiu technológiu než Čiarový kód. Jej hlavnými výhodami sú možnosť dekódovať obrázok len pomocou smartfónu s príslušnou aplikáciou a možnosť zakódovať väčšie množstvo informácií na rovnakú plochu v porovnaní s Čiarovým kódom. Navyše QR kódy dokážu zakódovať celú URL (Uniform Resource Locator), používa sa na presnú identifikáciu dokumentu na internete, čiže písmená a čísla, na rozdiel od čiarového kódu, ktorý dokázal zakódovať len čísla. Dokáže teda odkázať napríklad na webovú stránku, kde je opis tovaru. Často sa používa ako doplnok k pouličným plagátom, na overenie

lístkov na koncertoch, na označovanie krabíc v skladoch a samozrejme mnoho iných. Na obr. 1.2 je možné vidieť príklad QR kódu, ktorý odkazuje na autora tejto práce. [18]



Obr. 1.2: Príklad QR kódu

1.3 RFID tagy

RFID (Radio-Frequency Identification) tagy sa delia na Ultra High Frequency tag v preklade ultra vysokofrekvenčný tag, vyjadruje sa skratkou UHF tag, high frequency tag čo znamená v preklade ultra vysokofrekvenčný tag, skratkou vyjadrený ako HF tag a Low Frequency tag čo značí nízko-frekvenčné tagy v textoch sa značí skratkou LF tag. UHF, HF a LF tagy nenesú žiadnu informáciu používajú sa len na detekciu či sa v objekte nachádzajú alebo nie. Využívajú sa najmä v obchodoch na označenie ľahkoukradnuteľného tovaru, tag sa pripevní na oblečenie alebo prilepí na čokoládu. Táto technológia je jednoduchá, založená na jednoduchom obvode obsahujúcom cievku a kondenzátor, ktorý je možné detegovať v magnetickom poli. Existujú dva typy týchto systémov RF s rozkmitaním budiaceho elektromagnetického signálu jedná sa o jednoduchší a starší typ ďalším typom je RF s RFID čipom. RFID s čipom sa môžu deliť na podskupiny podľa toho či príslušnému tagu je možné informáciu prepísať alebo len čítať podľa toho sa delia na tagy, z ktorých je možné len čítať a na tagy z ktorých je možné čítať a podľa potreby aj zapisovať. [6]

1.3.1 LF tagy

LF tag (Low Frequency tag), jedná sa o tag, ktorý vysiela do okolia signál o frekvencii, ktorá zapadá do rozsahu nízkych frekvencií (125 kHz po 134 kHz). Tieto tagy majú dosah do 50 centimetrov, dokážu disponovať len menším objemom dát a sú pomerne drahé. Tento typ tagov sa používa na čipovanie zvierat, v dochádzkových systémoch alebo v imobilizéroch áut. [6]

1.3.2 HF tagy

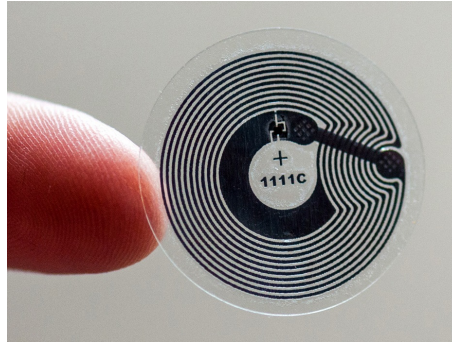
HF tag (High Frequency tag) je tag, ktorý vysiela frekvenciu v cca 13.56 MHz. Dosah tagov založených na vysokej frekvencii je pomerne krátky, informáciu z tagu je možné získať len do vzdialenosti menšej než meter od vysielača. HF tagy sa využívajú v bankovníctve na platobné karty (RFID čip). Najznámejší vysokofrekvenčný tag je NFC technológia (Near Field Communication).

RFID (Radio Frekvenčný Identifikátor) je čip implementujúci rádiové komunikačné rozhranie. Identifikátor obsahuje EPC identifikátor s dĺžkou väčšinou 96 bitov. Hodnotu identifikátora je možné nastaviť podľa potreby. Pokiaľ sa tento tag dostane do blízkosti antény, ktorá generuje elektromagnetické pole s rezonančným kmitočtom LC obvodu, tak sa v tagu naakumuluje energia, ktorá postačuje EPC čipu, aby vykonával svoju funkciu. Čip záťažovou moduláciou posunie anténe svoj 96 bitový identifikátor. Na prenos sa najčastejšie využíva pásmo 13,56 MHz. RFID tag sa dá deaktivovať silným signálom s frekvenciou rezonančného kmitočtu LC obvodu rovnako ako v RF tagoch s rozkmitaním budiaceho elektromagnetického signálu. Na kondenzátore sa vytvorí vysoké napätie, ktoré vďaka úmyselne zoslabeného dielektrika v kondenzátore spôsobí prieraz kondenzátora, čím sa tento tag stáva deaktivovaným. Jeho funkcia je nevratne ukončená. [6]

NFC (Near Field Communication), krátkodosahové vysokofrekvenčné bezdotykové spojenie na prenos dát medzi zariadeniami do vzdialenosti teoreticky až 4 cm, avšak reálne zatiaľ len do jedného centimetra. Využívaný v mobilných zariadeniach, napríklad smartfón, tablet a podobne. Ide o prenos dát podobne ako pri technológii Bluetooth len na krátke vzdialenosti. NFC čip obsahuje indukčnú cievku, ktorá vytvára elektromagnetické vlny, ktoré sú nutné na prenos energie. Súčasné NFC zariadenia majú len niekoľko centimetrov v priemere, čo je ich značnou výhodou. V porovnaní s Bluetoothom je NFC energeticky nenáročnejšia, lebo pre Bluetooth potrebuje napájanie prijímač aj vysielač. Pre NFC stačí, aby mal napájanie len prijímač. Prijímač vyšle elektromagnetické vlnenie čipu, ktorý vďaka cievke dokáže vlnenie prijať a premeniť na energiu, ktorá dostačuje čipu na odoslanie dát, ktorými disponuje. NFC môže fungovať samozrejme aj v móde, kde je napájaný prijímač aj vysielač. Takýmto spôsobom sa NFC používa najčastejšie na komunikáciu medzi dvoma telefónmi. Najčastejšie sa využíva pri bezkontaktných platbách pomocou telefónu, kde je výhodou, že prenos prebieha len na krátke vzdialenosti. Na obr 1.3 je možnosť vidieť NFC tag na fotografii.

1.3.3 UHF tag

UHF tag vysiela na veľmi krátkych vlnových dĺžkach, frekvencia týchto tagov je v rozmedzí 300 MHz až 3 GHz dosah až do 6 metrov. Jedná sa o jednorázovo



Obr. 1.3: Ukážka NFC tagu

programovateľný tag, informácia sa do tagu vloží pri výrobe. Tieto tagy umožňujú vysokú rýchlosť snímania. UHF tagy sa používajú na označenie batožiny na letiskách alebo na sledovanie paliet a kontajnerov.

1.4 Beacon

Beacon je malé zariadenie, ktoré opakovane vysiela do okolia signál, ktorý je možné prijať akýmkoľvek zariadením v dosahu s technológiou Bluetooth 4.0 Smart Ready. Bluetooth 4.0 vysiela na frekvencii 2,4 GHz a vie fungovať v dvoch rôznych režimoch:

- **zariadenie na periférii**

(teda konečné zariadenie), ktoré komunikuje len s jedným prvkom v lokálnej sieti, napríklad bezdrôtové príslušenstvá k počítaču:

- **centrálny režim**

, v ktorom funguje BLE v beaconoch. Je to režim, kde vysielať signál môže prijať viacero zariadení.

Signál s dátami, ktoré sú zapísané v čipe beaconu, sa vysiela do okolia každú desatinu sekundy. Beacon je možné umiestniť kdekoľvek vďaka gombíkovej batérii, ktorá, ho dokáže dotovať energiou až dva roky. Beaconu dokážeme nastaviť intenzitu vysielať signálu v dBm a periódu vysielať signálu v ms. Beacon je zariadenie, ktoré nevie nič o prostredí, v ktorom sa nachádza, ani o zariadeniach v okolí, má len jednu úlohu a to pravidelne vysielať rovnaký signál do okolia. Prenos dát medzi beaconom a prijímacím zariadením prebieha podľa definovaných pravidiel, čo znamená podľa komunikačného protokolu. Beacon môže podporovať komunikačný protokol iBeacon alebo Eddystone. Príklady rôznych beaconov sú k nahliadnutiu na obr. 1.4 [14, 9]



Obr. 1.4: Rôzne modely Beaconov

1.4.1 Rôzne druhy beacnov

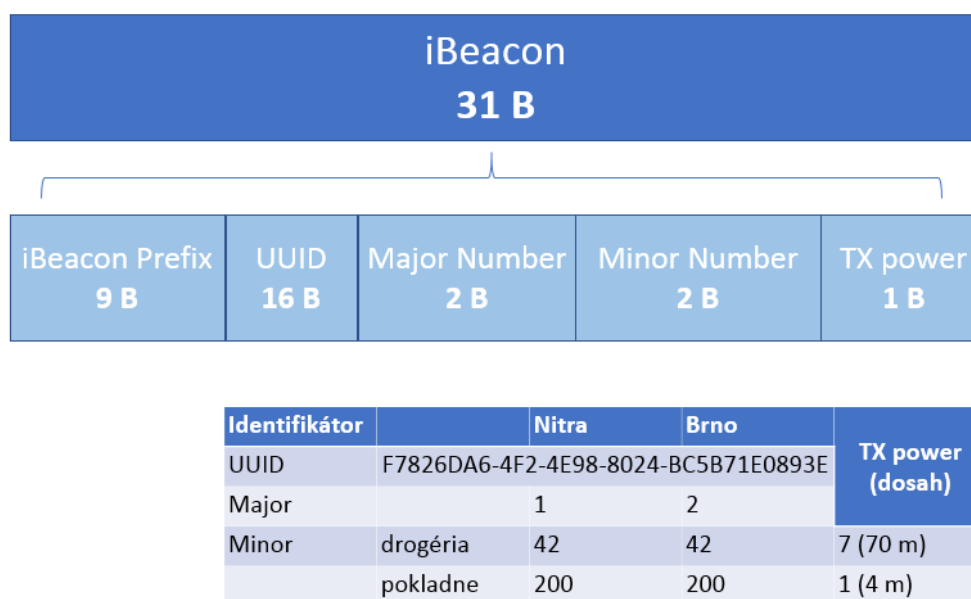
Na trhu sú rôzne druhy beaconov, ktoré disponujú aj inou technológiou než len Bluetooth 4.0. Niektoré drahšie beacny obsahujú technológiu NFC a RFID, Wi-Fi či pohybové alebo svetelné senzory. Niektoré z nich sú aj vodeodolné alebo konštrukčne odolnejšie voči rozbitiu. K drahším beaconom od Kontakt.io je možnosť dokúpiť si licenciu na API od Kontakt.io, pomocou ktorého je možné detegovať polohu užívateľa s presnosťou na centimetre. Táto licencia sa vzťahuje k jednému beaconu a platí sa 6 dolárov ročne.

1.4.2 iBeacon

iBeacon platforma bola prvý krát predstavená na konferencii Worldwide Conference 2013 spoločnosťou Apple v roku 2013 . Ide o komunikačný protokol, ktorý sprostredkúva komunikáciu medzi vysielateľom a prijímačom cez Bluetooth 4.0. Protokol je vyvíjaný a implementovaný spoločnosťou Apple a je súčasťou všetkých zariadení iOSu od verzie 7, patrí do Apple CoreLocation knižnice, čo je knižnica na určovanie polohy iOS zariadenia. Komunikácia cez BLE sa skladá z malých paketov, ktoré sa broadcastovo odosielaajú do okolia prostredníctvom rádiových vln. IBeacon technológia dokáže spolupracovať s GPS v systémoch s operačným systémom iOS. Táto technológia dokáže upozorniť prostredníctvom smartfónu na akciu opustenia alebo vkročenia do určitej lokality v okolí Beaconu [5, 12] .

Protokol iBeacon sa skladá z piatich blokov iBeacon prefix - 9 Bytes, UUID 16 Bytes, Major Number 2 Bytes, Minor Number 2 Bytes a TX power 1 Byte. Blok iBeacon Prefix sa skladá z ďalších blokov, ktorý prenáša údaje o majiteľovi zariadenia. Blok UUID prenáša id, ktoré sa väčšinou vzťahuje na konkrétnu aplikáciu. Viac beaconov môže mať rovnaké UUID, je to užitočné z toho dôvodu, že je možné mať viacero predajní rovnakého obchodného reťazca, ktorý bude komunikovať s jednou aplikáciou. Aplikácia detekuje v okolí UUID, z ktorého zistí či daný beacon

obsahuje pre ňu užitočnú informáciu alebo je to beacon k inej aplikácii. Ďalší blok Major Number je rozšírenie UUID a je možné pomocou neho lokalizovať beacon v predajni, všetky Beacons v predajni by mali priradené rovnaké Major Number. Nasledujúci blok, Minor Number je ďalším rozšírením pre Major Number a slúži na bližšie určenie polohy zákazníka v predajni. Každý Beacon v predajni má rovnaké Major Number avšak rôzne Minor Number. Z týchto troch blokov je možné zistiť, v okolí ktorého zariadenia sa užívateľ nachádza. Posledným blokom je TX power, v ňom je zapísaná hodnota intenzity signálu presne jeden meter od prístroja. Slúži na kalibráciu zariadenia, s touto hodnotou sa pracuje pri kalkulovaní odhadu vzdialenosti užívateľa od beaconu. Schematické zhrnutie blokov v protokole iBeacon je možné vidieť na 1.5 [12]



Obr. 1.5: Demonštrácia dátových blokov v protokole iBeacon [2]

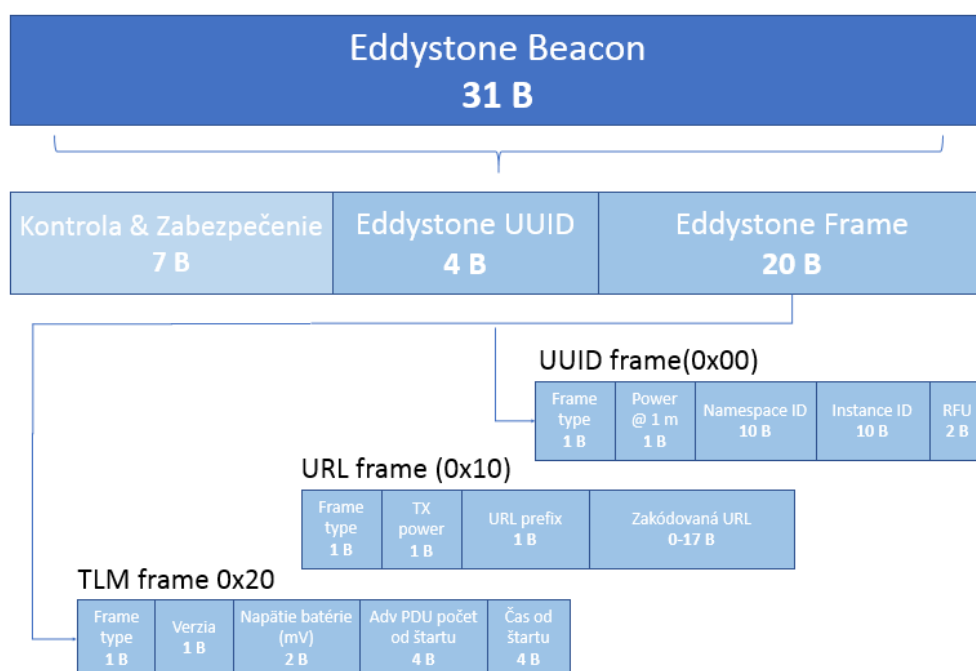
Lubovoľne editovateľné sú len bloky UUID, Major Number, Minor Number editovať ich je možné cez webové rozhranie firmy, od ktorej beacon pochádza. Avšak konečná konfigurácia samotného Beaconu musí prebehnúť cez zariadenie s technológiou BLE, ktoré má daný beacon v dosahu.

1.4.3 Eddystone

Eddystone platforma bola prvý krát predstavená v roku 2015 spoločnosťou Google. Tiež ide o komunikačný protokol, ktorý sprostredkúva komunikáciu medzi vysielateľom a prijímačom. Protokol Eddystone majú implementované všetky zariadenia s Androidom Lollipop (API 21) a vyššie. Protokol Eddystone sa tiež skladá z niekoľkých blokov:

- kontrola a zabezpečenie 7Bytes
- data 4 + 20 Bytes.

V pakete dát je odosielaná hlavička rámcu 4 Bytes a jeden z rámcov: Eddystone-UID sa skladá z rámcov Namespace a Instance Id, ktoré sú ekvivalentom UUID a Major z protokolu iBeacon, Eddystone-URL sa skladá z URL adresy avšak tento paket je obmedzený dĺžkou adresy, adresa nemôže presiahnuť 12 HEX znakov po zakódovaní. Tento rámec si vie prečítať ktokoľvek bez ohľadu na to, či má vašu aplikáciu alebo nie. Ďalším rámcom je Eddystone-TLM, v ktorom sú odosielané aktuálne telemetrické údaje zariadenia. Pomocou Eddystone EID vieme zabezpečiť, že z beaconu bude čítať len naša aplikácia. Schematické zhrnutie blokov v protokole Eddystone je možné vidieť na 1.6 [10, 11]



Obr. 1.6: Demonštrácia dátových blokov v protokole Eddystone [1]

V protokole Eddystone sú ľubovoľne editovateľné Eddystone-UID a Eddystone URL. Editovať ich je možné cez webové rozhranie firmy od ktorej beacon pochádza. Avšak konečná konfigurácia samotného Beaconu musí prebehnúť cez zariadenie s technológiou BLE, ktoré má daný beacon v dosahu.

1.4.4 Testované zariadenia

Beacon zariadenia využívané počas bakalárskej práce:

- **Kontakt.io**

podporuje protokol Eddystone aj iBeacon. Tento beacon podporuje navyše tagovanie jednotlivých zariadení. Jedno zariadenie môže mať maximálne dvadsať 32 znakových tagov. Po zmene tagov netreba cez Bluetooth nakonfigurovať beacon, ktorému bol tag zmenený, postačuje, keď sa zmeny uložia na cloude Kontaktu.io. Ak je prijímacie zariadenie (smartfón) pripojené na internet, tak vďaka základnému Software Development Kit (SDK) a Application programming interface (API) od kontakt.io je možné tag prečítať z ich cloudu. Smartfón zachytí uniqueId beacnu, toto id skúsi vyhľadať v cloude Kontakt.io medzi otagovanými zariadeniami, ak nájde zhodu do smartfónu vráti všetky tagy do ktorých daný beacon patrí. Dovoľuje zasiahnuť do intervalu odosielania signálu v rozmedzí od 20 ms do 10 240 ms. Umožňuje nastaviť intenzitu odosieleného signálu, rozdeliť to do ôsmich rôznych úrovní, jednotlivé úrovne sú prehľadne uvedené v tabuľke 1.1. Zvyšovaním úrovne výkonu sa skrátuje životnosť batérie. Optimálne nastavenie je interval odosielania dát na 350 ms a výkon na level 3 (dosah 20 metrov), na šetrenie batérie odporúčajú interval odosielania dát 633 ms a výkon znížiť na level 2 (dosah 10 m). Maximálny možný dosah udávajú na 70 m s predpokladom, že nie sú v okolí žiadne prekážky, ktoré by ovplyvnili šírenie signálu. Každý beacon obsahuje jednu 1000 mAh batériu, v kombinácii s úsporným Bluetoothom 4.0 dosahuje dĺžku životnosti tejto batérie 2 roky pri nastavení intervalu 350 ms a level výkonu 3, teda dosah 20 metrov.

Na stránkach firmy Kontakt.io je poskytnuté webové rozhranie komunikujúce s konfiguračnou aplikáciou v telefóne, ktorá je prepojená sériou overovacích kódov priamo k administrátorskému účtu a beacnov pripojených k nemu. Vo webovom rozhraní je umožnené nakonfigurovať daný beacon na iBeacon alebo Eddystone, poprípade oba naraz, priradiť rámcu hodnoty, v prípade Eddystone protokolu je možné rozhodnúť, ktorý sa bude a ktorý sa nebude odosielať. Eddystone totiž nepotrebuje odosielať všetky tri informačné rámce. Je možné odoslať akúkoľvek kombináciu rámcov Eddystone-UID, Eddystone URL a Eddystone TLM. Po uložení zmien uskutočnených vo webovom rozhraní, aplikácii v smartfóne príde notifikácia na konfiguračnú aplikáciu a jednoduchým potvrdením synchronizácie je beacon nakonfigurovaný v prípade, že je v dosahu a príslušný účet má práva na zmeny odosielených informácií v zariadení. Tieto zariadenia je možné vidieť na obr. 1.7.

- **Proxiot**

beacon podporuje Eddystone aj iBeacon rámec, tento beacon je možné nastaviť na štyri rôzne výkonnostné úrovne 4 dBm, 0 dBm, -6 dBm, -23 dBm. Interval vysielania je možné nastaviť v rozpätí 100 ms až 10 sekund.

- **iBKS**

podporuje odosielanie iBeacon aj Eddystone protokolu, k dispozícii je len konfiguračná aplikácia bez webového rozhrania. iBKS beacon podporuje osem výkonnostných úrovní -30 dBm, -20 dBm, -16 dBm, -12 dBm, -8 dBm, -4 dBm, 0 dBm, 4 dBm. Porovnanie beacnov, ktoré sú k dispozícii je zhrnuté v tabuľke 1.2.

Tab. 1.1: Možnosti upravenia výkonu Kontakt.io beaconu

Level výkonu	ddecimal value	Výkon signálu 1 meter	Dosah
0	-30 dBm	-115 dBm	2m
1	-20 dBm	-84 dBm	4m
2	-16 dBm	-81 dBm	10m
3	-12 dBm	-77 dBm	20m
4	-8 dBm	-72 dBm	30m
5	-4 dBm	-69 dBm	40m
6	0 dBm	-65 dBm	60m
7	4 dBm	-59 dBm	70m

Tab. 1.2: Porovnanie beacnov rôznych výrobcov

výrobca	úrovne výkonu	webové rozhranie	podpora tagov	spokojnosť s aplikáciou [%]	spokojnosť s podporou [%]	cena za kus [€]
Kontakt.io	8	áno	áno	100	90%	30 - 120
PROXIoT	4	nie	nie	60	90%	16
iBKS	8	nie	nie	0	0%	16

1.4.5 Možnosti využitia technológie v reálnom svete

Aj napriek tomu že beacon technológia je pomerne nová našla si množstvo spôsobov využitia v rôznych odvetviach.

Marketing

Je tu možnosť použitia beacnov v obchodných centrách, kde by mala predajňa svoj beacon a v okamihu, keď kupujúci vkročí do dosahu beaconu v predajni, tak si smartfón vie prečítať údaje a odoslať na web, ktorý vie jednoznačne identifikovať beacon, z ktorého správa pochádza. Na základe zistených informácií môže web vygenerovať nejakú zľavovú ponuku, ktorú zobrazí užívateľovi a ten ju môže okamžite využiť v predajni, pri ktorej sa nachádza.



Obr. 1.7: Fotka testovaných zariadení od Kontakt.io (Eddystone iBeacon), Proxiot (Eddystone iBeacon), iBKS (iBeacon)

Do rozsiahlejších predajní je možné implementovať beacons s inou logikou. V predajni je mnoho rôznych druhov tovaru, pričom nad každým druhom tovaru (elektronika, nábytok, oblečenie, ...) môže byť situovaný beacon. Ďalší beacon by bol nad pokladňami a posledný by bol umiestnený nad východ. Návštevník príde prechádzať sa pri rôznych regáloch, ktoré ho zaregistrujú a pomocou aplikácie v smartfóne o pohybe užívateľa informuje server, ktorý má prehľad o tom, v ktorom oddelení návštevník strávil najviac času. Návštevník si nič nevybral a odchádza z predajne, kde ho zaregistruje beacon nad východom z predajne. Znovu pomocou aplikácie v telefóne je o odchode informovaný server, ktorý na základe informácií o čase strávenom pri jednotlivých regáloch je schopný vygenerovať ponuku, ktorú by návštevník mohol oceniť. Po jej prečítaní ju môže ísť využiť, vrátiť sa do predajne. Pri platení produktu musí prejsť okolo pokladní, kde ho zaregistruje beacon nad pokladňami. Server je informovaný, že návštevník prešiel okolo pokladní a pri odchode mu nebude poslaná žiadna ponuka, lebo si už niečo kúpil.

Podobne by sa dal využiť na verejných podujatiach a užívateľom, ktorí navštívili koncert/prezentáciu/prednášku, je možné poslať nejaké dodatočné informácie alebo limitovanú ponuku nejakého tovaru.

V Českej republike beacons používajú spoločnosti OBI vo viacerých mestách a Obchodný dom Kotva v Prahe na navigáciu a odovzdávanie informácií zákazníkom.

Lokalizácia

Zariadenie je možné použiť na pripomienky v mestskej hromadnej doprave. Beacons sú v okolí zastávok, každá zastávka má svoje unikátne id alebo je zaradené do

jedinečného tagu. Keď sa smartfón dostane do okolia beaconu s požadovaným id alebo tagom tak, aplikácia môže zobrazíť upozornenie, že sa blíži zastávka, na ktorej užívateľ plánoval vystúpiť. Niečo podobné sa využíva v Londýne. [7]

Pomocou beacon zariadení sa dá pomerne presne sledovať poloha užívateľa v budovách, kde GPS často zlyháva. Presné určovanie polohy v budovách je možné využiť v múzeách alebo na výstaviskách, kde je možné navigovať používateľa priamo tam, kam chce ísť, poprípade po dostatočnom priblížení k exponátu sa môže na zariadení automaticky spustiť prezentačné video alebo len zobrazíť popis. [7] Pomocou technológie beacon je možné vytvoriť podobnú hru ako pokemon GO, s tým rozdielom, že oklamanie signálu z beaconov, ktoré zisťujú polohu užívateľa sa javí zdanlivo nemožné na rozdiel od GPS ktorej poloha sa dá jednoducho simulovať.

Spoločnosť American Airlines testuje beacony na letisku v Dallase. Pomocou zariadení pripevnených po celom objekte letiska sa zákazníci dokážu lepšie orientovať na letisku a vedia si pohodlne nájsť cestu k bezpečnostným prepážkam, informáciám či občerstveniu. Ďalšou z možností využitia tejto technológie na verejných priestranstvách sa javí využitie na športových štadiónoch, kde po zadaní zakúpeného sedadla vás aplikácia vie bezpečne navigovať k vášmu miestu. [7]

Zabezpečenie

Pomocou beaconov by mohlo byť bezpečnejšie odomykanie dverí, či už auta alebo domu. V aute alebo v dome by bol umiestnený beacon, ktorý je schopný kontrolovať, či je v okolí práve váš telefón a na základe kontroly dokáže umožniť odomknutie dverí skrz aplikáciu v telefóne. Na to, samozrejme, musí byť prispôbená technológia aj vo dverách. [7]

Spoločnosť Starwood hotels v niektorých svojich hoteloch používa chytrú recepciu, kde sa návštevník zaregistruje v aplikácii, sám sa odnaviguje k svojmu apartmánu a tam si poklepaním na displej telefónu sprístupní svoj apartmán a to všetko bez použitia kľúča. [4, 7]

Dohľad

Zamestnávatelia by mohli namiesto prístupových kartičiek používať beacon nad vchodom a aplikáciu k nemu príslušnú. Zamestnanec vchádza do práce okolo beaconu a ako odpoveď servera môže byť otvorenie dverí. Týmto spôsobom by zamestnávateľ mal prehľad o tom, ako dlho zamestnanec bol v práci.

2 TECHNOLOGIE POUŽITÉ V APLIKÁCI

Táto časť sa bude vysvetľovať aká technológia alebo software bude použitý vo finálnej aplikácii a prečo.

2.1 Android studio

Aplikácia bude bežať na smartfónoch s operačným systémom Android od verzie Lollipop 5.0 (API 23). Vývoj aplikácie prebehne v prostredí Android studio 3.0.0+, ktoré je vyvíjané vývojármi softwarov JetBrains. Je to pravidelne aktualizované prostredie. Každým aktualizovaním pribudnú nové knižnice na ľahšiu tvorbu responzívnych layoutov, zefektívnia prácu dávnejšie vytvorených knižníc, poprípade opravujú chyby, ktoré vznikli v predošlom vydaní prostredia. Od verzie Android studia 3.0.0 je podporovaný paralelný vývoj Android aplikácií v jazyku Java a Kotlin. Programovací jazyk Kotlin je dielom softvérového vývojára JetBrains, jedná sa o jazyk prispôbený na tvorbu Android aplikácií. Vychádza z Javy 8.0, v ktorej je množstvo zjednodušení a efektívnejších algoritmov než v starších verziách. V Android štúdiu je možné používať v kotlin projekte Javu aj Kotlin, to znamená, že je možné používať programovací jazyk Java aj Kotlin v jednom projekte, avšak treba upozorniť že toto nefunguje naopak, nie je možné používať Kotlin v Javovskom projekte. Na druhú stranu je možné používať oba typy tried v projekte. Ak niekto zverejnil riešenie nejakého problému šikovne a efektívne, je možné v projekte vytvoriť kotlinovskú triedu, do nej skopírovať danú metódu a zvyšok triedy písať v Jave. Android studio má grafické prostredie na tvorbu layoutov, grafika aplikácii sa tvorí pomocou značkovacieho jazyku XML. Aplikácie je možné spustiť priamo z prostredia a to buď pomocou AVD (Android Virtual Device) v tomto simulátore sa dá stiahnuť nejaký z rodiny Nexusov ďalej je možné vybrať si z množstva pluginov na virtuálne smartfóny, napríklad Genymotion, v ktorom je na stiahnutie množstvo samsungov, htc, nexusov. V Genymotion je možné stiahnuť emulátor na Android TV a smartwatch. Ďalšou možnosťou testovania aplikácie pomocou Android studia je pripojenie reálneho telefónu pomocou ADB (Android Debug Bridge), cez ktorý je možné spustiť aplikáciu v hardwarovom telefóne cez Android studio. V každej z týchto možností testovania aplikácii popri vývoji je k dispozícii možnosť debugovania. Debugovanie v Android studio je jednoduché, jednoducho je možné zistiť hodnoty premenných, zložitejšie premenné napríklad triedy je možné si rozkliknúť a pozrieť vnorené premenné.

Používanie obrázkov je možné buď priamo napísaním kódu obrázku pomocou XML, poprípade vektoru alebo rastrového obrázku vo viacerých rozlíšeniach. Každý typ súboru je uložený vo vlastnom priečinku. **Res**, čo je skratka z resources, priečinok

je priečink obsahujúci všetku grafiku a súbory len na čítanie. Pre lepšie vysvetlenie: rastrové obrázky s príponou jpg, png, jpeg, gif sa ukladajú do priečinka **drawable**, ktorý sa nachádza v priečinku **res**, do tohto priečinku sa ukladajú aj už importované vektorové obrázky, a to ako súbor s príponou .xml. Tento súbor nie je čitateľný po otvorení cez počítač, lebo obrázok je vyjadrený v sekvencii znakov. textové súbory na čítanie väčšinou s príponou .txt sú situované do priečinku **raw**, ktorý sa tiež nachádza v priečinku **res**. Ďalším typom súborov sú súbory s príponou .ttf čo je prípona pre vlastné štýly písma. V Android studiu sú importované fonty z rodiny Roboto, čo je verejne prístupný štýl písma od Googlu, tieto súbory sa ukladajú do priečinku **assets** tiež v priečinku **res**.

2.2 MySQL

MySQL je otvorený viacvláknový databázový server. MySQL je priamo implementovaný vo viacerých programovacích jazykoch ako napríklad Perl, C++ a PHP, pomocou ktorého bude aplikácia komunikovať s MySQL serverom. Samotný MySQL server môže obsahovať viacero tabuliek, ktoré môžu byť navzájom rôzne poprepájané. Prezeranie tabuliek uložených na MySQL serveri je možné pomocou webového rozhrania na to určeného, toto webové rozhranie je poskytnuté poskytovateľom tejto služby. Pre aplikácie spomenuté v tejto práci pristupujeme k tabuľkám cez poskytovateľa Wedos [15]. V bakalárskej práci sa používa verzia s označením 8.0.

2.3 PHP

Hypertext Preprocessor známejší pod skratkou PHP. Jedná sa o populárny open source skriptovací jazyk, ktorý je používaný najmä na komunikáciu klient-server aplikácií na strane servera. PHP dokáže spolupracovať s relačnými databázami ako napríklad MySQL, Oracle, Microsoft SQL server, SQLite. Je schopný bežať na takmer všetkých operačných systémoch vrátane UNIXu, Linuxu, Windows aj Mac OS X. [16] V bakalárskej práci sa používa verzia s označením 3.8.0.

2.4 JSON

JavaScript Object Notation je odľahčený formát pre výmenu dát. Je jednoducho čitateľný, jednoducho spracovateľný a nenáročný aj na písanie človekom. JSON je textový formát využívajúci konvencie z vyšších programovacích jazykov, preto je JSON ideálny na výmenu dát medzi platformami. [13] V bakalárskej práci sa využíva JSON na komunikáciu medzi databázou a Android smartfónom.

2.5 Firebase

Firebase je mobilná a webová developerská platforma kúpená spoločnosťou Google v roku 2014. Firebase poskytuje množstvo užitočných funkcií ako je realtime databáza, uložisko, webový hosting, testovanie pre Android aplikácie, dynamické linky, správa reklám, notifikácie, remote config a mnoho ďalšieho. V našej aplikácii bude od Firebase používaná možnosť odosielania a prijímania push notifikácií. Push notifikácia je notifikácia v smartfóne, ktorej prijatím sa prebudí aplikácia, ku ktorej patrí. Tá spracuje prijaté dáta a vytvorí notifikáciu, ktorú ukáže užívateľovi. Tieto notifikácie sa dajú poslať jednému konkrétnemu užívateľovi alebo všetkým užívateľom naraz. Mobilné zariadenie užívateľa sa identifikuje tokenom uloženým v MySQL databáze. Tento token je nutné uložiť pri prvom spustení, respektíve prihlásení užívateľa do danej aplikácie. Token sa vzťahuje k zariadeniu, z ktorého je užívateľ prihlásený, nie priamo k nemu. Všetky tieto veci sa jednoducho implementujú importom vhodného pluginu [8].

2.6 Realm

Pre lokálne uloženie informácií potrebných na chod Android aplikácie budú tieto informácie uložené pomocou platformy Realm. Realm slúži na uloženie dát na zariadenie užívateľa, niečo podobné ako serializácia alebo parcelovanie, len omnoho efektívnejšie a rýchlejšie. Možnosťou by bolo využitie lokálnej SQL databázy, ktorá je implementovaná do Android systému už dávno, avšak pre väčšie množstvo dát je Realm rýchlejší v zápise, čítaní aj mazaní. Pre účely tejto práce nebude aplikácia pracovať s takým množstvom dát aby bol citelný rozdiel medzi Realmom a SQL. Postupom času budú všetky databázové systémy prechádzať na NOSQL databázu, ktorej príkladom je Realm. Takže výsledný modul bude na to pripravený a nebude treba prerábať komunikačnú logiku na veľa miestach. V prípade vynechania MySQL databáz na strane serveru by bolo treba len zmeniť synchronizáciu sťahovania dát z webu, čo nie je žiaden zásah do logiky beacnov. [17]

3 NÁVRH A IMPLEMENTÁCIA APLIKÁCIÍ

V tejto sekcii je bližšie opísaná aplikácia a všetky potrebné komponenty na jej bezproblémový beh.

3.1 Meranie intenzity vysielaného výkonu

Na lepšie pochopenie správania beacnov v reálnom prostredí bolo uskutočnené meranie prijatého výkonu z beacnu v určitých vzdialenostiach od neho. Účelom merania bolo zistiť intenzitu signálu v rôznych vzdialenostiach od vysielacích beacnov. Dôležité je zistiť, či intenzita prijatého signálu konštantne klesá so vzdialenosťou od beacnu alebo je prijatý signál ovplyvnený interferenciou. Najväčší výrobcom udávaný dosah beacnov je okolo 70 metrov. Všetky tri značky beacnov podporujú protokol iBeacon, ktorý bol v tomto meraní použitý na každom z nich. Cieľom bolo zmerať intenzity prijatého výkonu z každého meraného bodu aspoň stokrát, potom zo všetkých meraní je vybratá jedna hodnota, ktorá reprezentuje nameranú hodnotu v danej vzdialenosti. Vlastnosť reprezentujúcej hodnoty je, že je najväčšia spomedzi všetkých meraní v rovnakej vzdialenosti. V testovacích podmienkach trvalo zmeranie 100 hodnôt v bezprostrednej vzdialenosti od beacnu vyše 10 minút, preto meranie bolo uskutočnené paralelne na troch zariadeniach s operačným systémom Android Xiaomi Mi3 s Androidom 6.0, Xiaomi Mi6 s Androidom 8.0 a Lenovo Tab3 plus s Androidom 6.0.

Scenár pre meranie výkonu

Najlepšia možnosť bola odoslať namerané dáta na server. Výhoda odosielania údajov na server je v tom, že je možné odoslať viac údajov než len intenzitu signálu v danom mieste, avšak je treba odosielať nejakú referenčnú vzdialenosť aby bolo možné z údajov zistiť, z akej pozície je údaj odoslaný. Referenčná vzdialenosť bude zadávaná užívateľom, pre jednoduchšie ovládanie referenčná vzdialenosť sa bude zväčšovať o jeden meter vždy po stlačení tlačidla na displeji, aby nebolo potrebné na každom meranom stanovisku stláčať tlačidlo na synchronizáciu s beacnami a odoslanie.

Implementácia

Je vhodné do aplikácie implementovať SDK od spoločnosti Kontakt.io, pomocou ktorého je možné jednoduchšie detegovať a prečítať informácie z beacnov, najlepšia možnosť bude využiť listenery z balíčku, ktoré sú schopné vždy pri zmene stavu beaconu vyvolať nejaký blok kódu. Stavby beacnov sú:

- Discovered je volaná pri nájdení nového beacnu,

- Updated je volaná pri opätovnom nájdení beacnov v dosahu
- Lost je volaná niekedy keď sa daný beacon stratí z dosahu, nemusí sa zavolať u každého beacnu

To znamená, že meranie bude prebiehať automaticky. Jediné čo užívateľ musí spraviť, je pri spustení aplikácie tlačidlom spustiť meranie, počas merania je treba pravidelne zvyšovať referenčnú vzdialenosť od vysieláčov. Vizuálny náhľad aplikácie viď Obr. 3.1. Keďže bolo definovaných 100 vzoriek z každého bodu merania a meranie prebiehalo paralelne na troch zariadeniach, tak aplikácia indikovala zmenu farebnej škály na príslušný meraný beacon, počet zmeraných vzorkov. Screen z aplikácie je k nahliadnutiu na obrázku 3.1.

- **Červená** 0-25 vzoriek
- **Oranžová** 26-40 vzoriek
- **Tmavo zelená** 40-60 vzoriek
- **Zelená** 100 a viac vzoriek

Ideálny prípad merania na stanovišti je ten, že na všetkých meracích zariadeniach majú indikátory počtu nameraných vzoriek zelené farby, vtedy je možné presunúť meracie stanovište na iné miesto. Avšak v reálnom prostredí merania trvali moc dlho na to aby bolo sto výsledkov z každého zariadenia, tak sa meralo pokiaľ nebol zdanlivý súčet z indikátorov väčší než sto. Pre Beacon od spoločnosti Proxiot nebolo možné zmerať sto vzoriek z jedného metra za 20 minút, ale len približne 50, tak sa meranie upriamovalo hlavne na Kontakt a IBKS beacny. Ktoré vzdalovaním sa od beacnov naberali aj na čase potrebnom na zmeranie sto vzorkov v súčte. Od vzdialenosti 37 metrov sa znížil požadovaný počet vzorkov na 50. Počas merania bol zistený fakt, že na tablete boli dosiahnuté indikačné hodnoty ako prvé a omnoho skôr než na ostatných zariadeniach. Xiaomi Mi3 prestalo detegovať signál vo vzdialenosti 44 metrov.

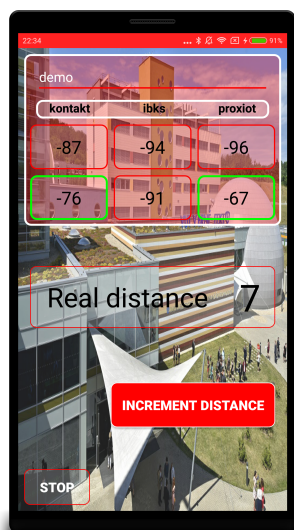
Postupnosť komunikácie používanej pri meraní je zobrazená v sekvenčnom diagrame 3.2. V komunikácii figurujú tri prvky, beacon, smartfón a databáza (do databázy sa vkladajú informácie pomocou php skriptu).

1. **Beacon:** v pravidelných intervaloch vysieľa do okolia informácie, ktorými disponuje. Tieto informácie sú prijaté v smartfóne.

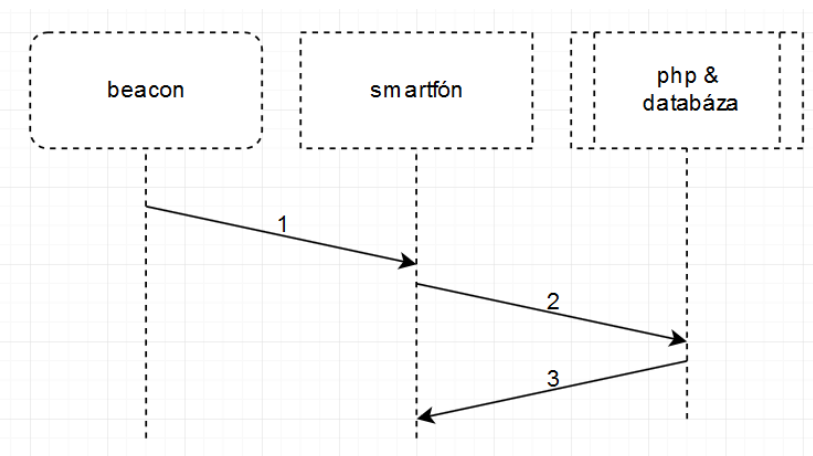
2. **Smartfón:** informácie prijme a spustí komunikáciu s databázou, začne sa odosielanie/ukladanie dát.

3. **Databáza:** po úspešnom uložení dát je odosielateľ (smartfón) informovaný o dokončení transakcie. Aplikácia informuje užívateľa po sto úspešne odoslaných transakciách zafarbením políčka na zeleno.

Dáta odosielané na server sú nasledovné: referenčná vzdialenosť, výkon prijatého signálu, vzdialenosť od vysieláča odhadnutú pomocou Kontat.io SDK, unikátne ID



Obr. 3.1:
Vizuálna ukážka
aplikácie na me-
ranie intenzity
signálu

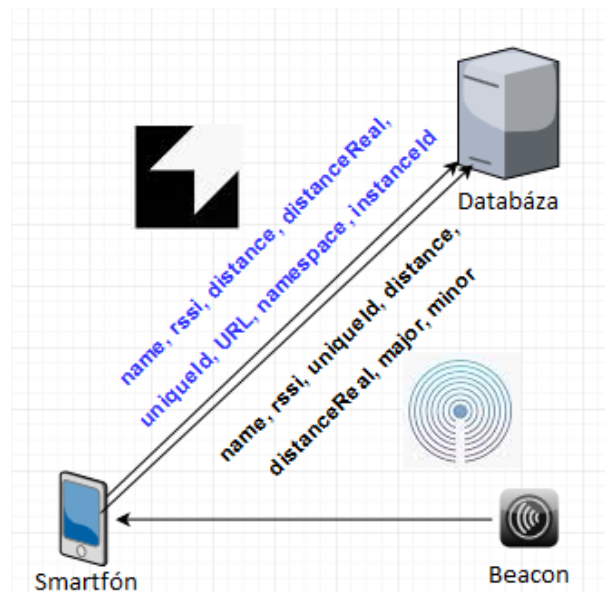


Obr. 3.2: Zobrazenie postupnosti komunikácie medzi bea-
nom, telefónom a databázou

a aktuálne nastavenú hodnotu výkonu beaconu. Ďalšie odosielané informácie závisia od použitého protokolu Eddystone alebo iBeacon. V prípade Eddystone protokolu je možné odosielať informácie z rámcov Eddystone URL a Eddystone UID 1.4.3 , ktorý obsahuje namespace a instanceId. V prípade iBeacon protokolu je možné odosielať na server informácie UUID 1.4.2 a major a minor čísla. Pre jednoznačnejšiu identifikáciu nameraných údajov na serveri bude zavedená textová premenná vyjadrujúca, ktoré meranie práve prebieha, napríklad "prostredie 1 TxPower0 bok“. Túto premennú užívateľ bude musieť pred spustením merania vyplniť, pri spracovávaní merania to ušetrí množstvo času, keďže cez phpadmina je možné vyexportovať všetky údaje splňujúce určitú podmienku. Je pravda, že medzi odosielanými informáciami na server bola spomenutá aj informácia o aktuálne nastavenom výkonnostnom leveli avšak, nie je vhodné spoliehať sa na niečo, čo nie je empiricky otestované. Schématické vysvetlenie komunikácie je možné vidieť na obr. 3.3. Na serveri sa plnili dve tabuľky jedna pre Eddystone protokol a druhá pre iBeacon. Na obrázku 3.3 sú naznačené aké dáta môžu byť ukladané v závislosti od typu protokolu.

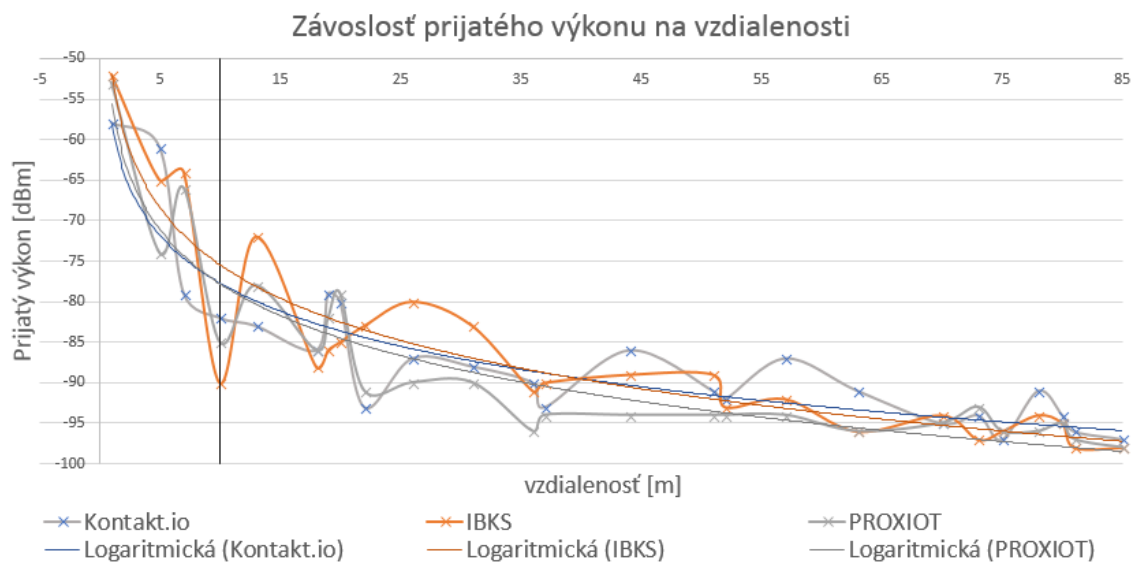
3.1.1 Vyhodnotenie merania

Táto kapitola sa zaoberá spracovaním nameraných údajov, vyobrazením výsledného grafu a zhrnutím výsledkov merania.



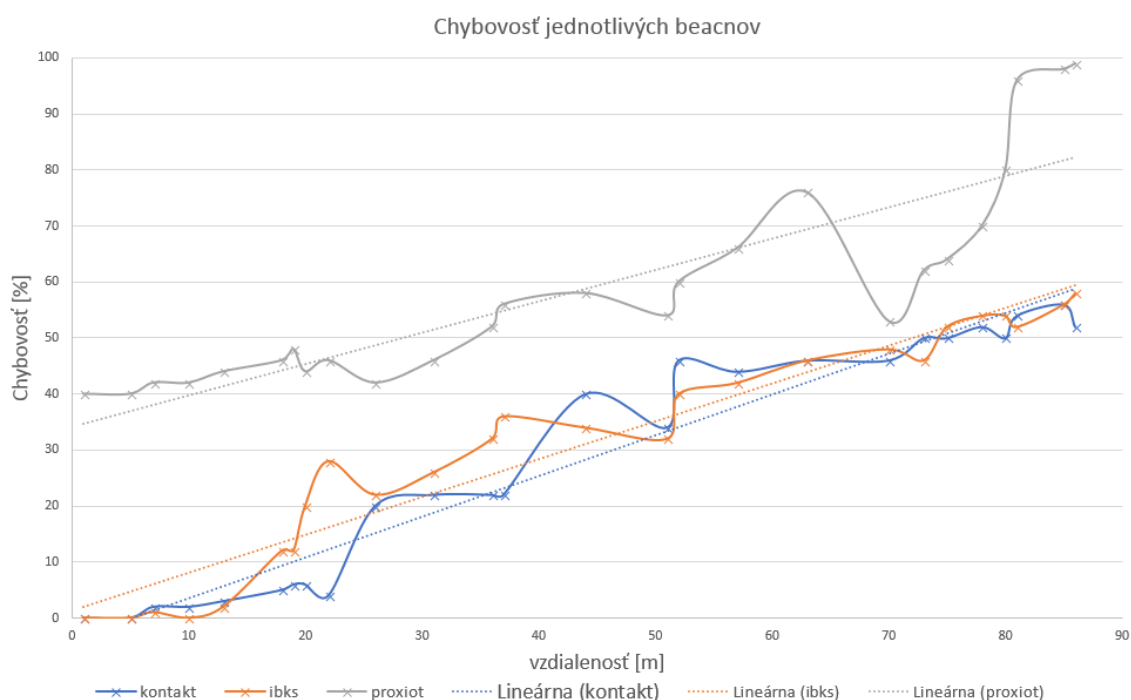
Obr. 3.3: Diagram opisujúci komunikáciu beacon -> smartfón -> server pri meraní

Po meraní bolo v databáze 3457 výsledkov z 27 meracích bodov troch rôznych výrobcov beacnov: Kontakt.io, PROXIoT, IBKS. Z týchto hodnôt je nutné vybrať 27 reprezentačných bodov pre všetky tri beacny. Reprezentačné body sa vybrali funkciou ktorá určí najväčší prvok z jednotlivých meraní v určitej vzdialenosti. Dôvod vybratia maximálnej hodnoty a nie modusu je ten, že hľadáme najlepšiu možnú hodnotu prijatého výkonu. Rozdiel medzi maximálnou a minimálnou hodnotou v jednotlivých vzdialenostiach bol 2-6 dB. Reprezentačné hodnoty z každého beacnu sú vyobrazené na grafe 3.4



Obr. 3.4: Závislosť prijatého výkonu vzhľadom na vzdialenosť

Z obrázku 3.4 je možné výčítať, že najstrmšia strata prijatého výkonu pre všetky beacny je na prvých desiatich metroch. Na zvyšku meraného intervalu je graf všetkých beacnov nemonotónny a prijatá intenzita signálu neklesá pravidelne. Sú úseky kde dochádza k nečakane vysokej hodnote prijatého signálu. Čo sa týka chybovosti prijatých a spracovaných dát tak je nulová, čo znamená že počas celého merania boli doručené validne dáta z beacnu. Avšak nie každá prijatá správa bola úspešne spracovaná, viacero sa ich stratilo, predpoklad je, že to je spôsobené chybou v štruktúre prijatého paketu, čím bolo zapríčinené, že ho nebolo možné spracovať. Každý beacon mal nastavenú frekvenciu odosielania správ na dvadsať správ za sekundu, čo znamená že každú sekundu odoslal beacon do okolia 20 rovnakých správ. Z toho vyplýva, že za päť sekúnd by mal smartfón obdržať sto vzoriek. Zariadenia vo väčších vzdialenostiach prijali menší počet správ než 100. Závislosť tejto vážnej chybovosti je zobrazená v grafe 3.5.



Obr. 3.5: Chybovosť prijatých správ z beacnu vzhľadom na vzdialenosť od beacnu

Týmto meraním sme zistili, že sa nedá spoliehať na výrobcom udávaný dosah na výkonnostnom leveli. Všetky beacny boli nastavené aby dosiahli približnú maximálnu vzdialenosť 70 metrov. Z grafu je očividné, že všetky beacny túto vzdialenosť bez problémov prekonal. Z toho vyplýva, že najrozumnejšie je kontrolovať prijatý výkon z beacnu a na základe toho nastaviť logiku priblíženia k beacnu. To znamená, že logiku v aplikácii by nebolo rozumné postaviť na tom, že ak zdetekujem signál z beacnu B, tak to znamená, že som od neho vzdialený menej než 70 metrov, za

predpokladu, že je na výkonnostnej úrovni s dosahom max 70 metrov, treba logiku aplikácie postaviť na tom, že pozrie s akou intenzitou zariadenie obdržalo správu z beacnu a ak je tá intenzita väčšia než -90 dBm tak je beacon vo vzdialenosti menšej než 70 metrov od beacnu. Táto logika tiež nemusí byť stopercentná, ale určite bude spoľahlivejšia.

3.2 Aplikácia zastrešujúca marketing v obchodoch

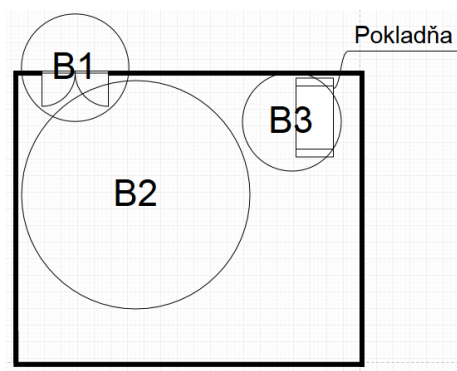
Návrh aplikácie s marketingovou logikou pomocou technológie beacnov. Táto aplikácia je kompatibilná s operačným systémom Android od verzie 5.0. Aplikácia je navrhnutá na cielený marketing na základe pohybu užívateľa v predajni. Na lokalizáciu užívateľa v objekte sa používa beacon technológia. Aplikácia rozhoduje akú notifikáciu odošle a odošle notifikáciu na základe informácií o pohybe užívateľa.

3.2.1 Testovací scenár

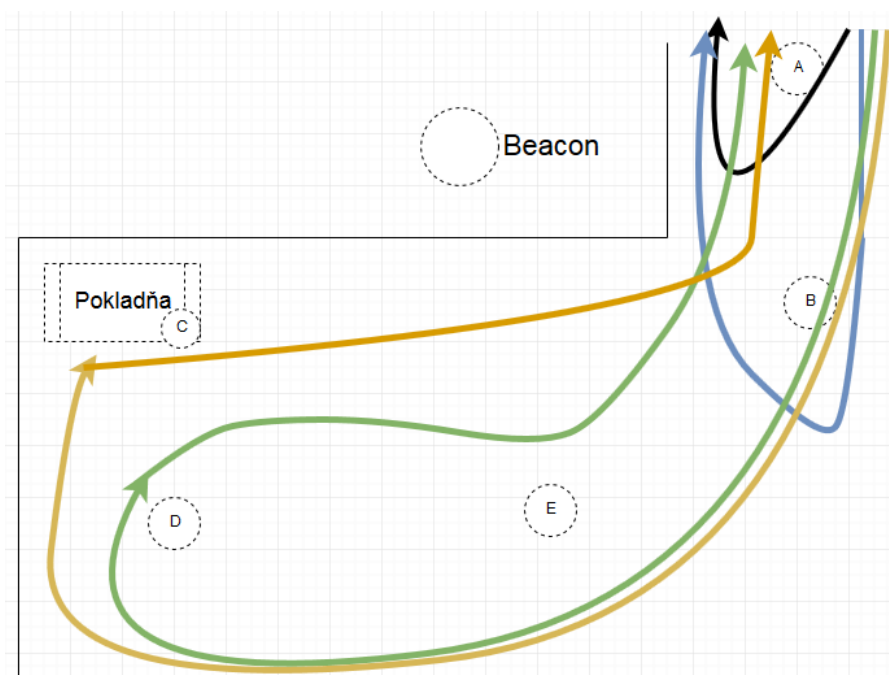
Rozvrhnutie beacnov do prostredia je nasledovné: jeden beacon nakonfigurovaný na veľký dosah bude situovaný nad vchod obchodu, označovaný ako BEACON1. Ďalší bude situovaný do okolia pokladní, s nastavením nech je detegovateľný každým zariadením, ktoré prejde cez pokladne (zaplatí), označený ako BEACON3. Predpoklad je taký, že vchod do obchodu slúži zároveň aj ako východ. Tretí beacon je umiestnený do časti obchodu, okolo ktorej musí návštevník vždy pri vstupe prejsť, ďalej len BEACON2. Pre bližšiu predstavu je rozvrhnutie beacnov v obchode zobrazené na obr. 3.6. Na obrázku 3.7 sú znázornené rôzne možnosti pohybu užívateľa v predajni.

- V prípade, že by užívateľ mal trajektóriu pohybu ako je znázornená **čiernou** šípkou, návštevník by nedostal žiadnu notifikáciu pri odchode a to z dôvodu, že neprejavil záujem a nevšiel do predajne hlbšie (nezaznamenal ho beacon B).
- V prípade **modrej** trajektórie by pri odchode užívateľ notifikáciu obdržal.
- **Zelená** varianta vyobrazuje scenár kedy sa užívateľ motá po predajni avšak na koniec si nič nekúpi. Pri odchode mu bude doručená notifikácia so zľavou na tovar, ktorý sa nachádza v okolí beacnu D alebo E, v závislosti od toho pri ktorom Beacne sa užívateľ zdržal dlhšie.
- Posledná trajektória na obrázku je **žltá**, užívateľ sa prešiel po predajni vybral si tovar, ktorý na pokladni zaplatil a odchádza. Pri odchode mu nebude doručená žiadna notifikácia.

Keď príde užívateľ do obchodu, zachytí ho BEACON1 **A** pri vchode, na server sa odošle informácia, že užívateľ prešiel okolo beacnu nad vchodom. Ďalší očakávaný



Obr. 3.6: Názorné rozmiestnenie beaconov v jednoduchom obchode



Obr. 3.7: Možnosti pohybu návštevníka

signál servera bude detekcia vysieláča BEACON2 **B**, ktorým server dostáva potvrdené, že užívateľ naozaj vošiel do predajne, čiže sa neotočil hneď po vkročení do obchodu. Ďalšia očakávaná informácia o pohybe užívateľa príde, ak užívateľ prejde okolo pokladní, kde zachytí signál od BEACNU3 **C**. Pomocou neho sa na serveri k užívateľovi zapíše informácia o prechode okolo pokladní, čo znamená, že užívateľ zaplatil za nejaký kúpený produkt. Ďalšie kroky užívateľa budú k východu/vchodu kde je znovu prijímaný signál z BEACONU1 **A**, ktorý vyšle na server otázku, či užívateľ niečo kúpil alebo nie. Ak nie, tak bude priamo na jeho telefón vygenerovaná notifikácia s nejakým zlavovým kupónom na nejaký tovar. Beacony E a D slúžia na cielenejšie sledovanie pohybu užívateľa a pomocou nich sa zisťuje, pri ktorom regále

bol návštevník dlhšie, ktorý ho viac zaujal. Napríklad návštevník prišiel do obchodu za účelom pozrieť koľko stojí kniha, ktorú hľadá, lebo v rámci hľadania najlepšej ceny obzerá viaceré obchody. Najviac času strávi pri knihách beacon **D**, ale ako to býva, tak sa prejde aj po zvyšku obchodu. Nič si nekúpi, takže neprejde okolo pokladní, z čoho vyplýva, že pri odchode z obchodu server zistí, že užívateľ nič nekúpil a vygeneruje cielenejšiu notifikáciu v podobe desaťpercentnej zľavy na knihy. Užívateľ si premyslí, či zľavu využije alebo nie. Ak áno, tento proces sa zopakuje s rozdielom, že tentokrát pri odchode z predajne nebude na jeho smartfón adresovaná žiadna notifikácia, keďže si nejaký produkt v obchode kúpil.

3.2.2 MySQL Databáza

V tejto sekcii sú bližšie informácie k tabuľkám nutným k bezproblémovému chodu aplikácie. Tabuľka je zoskupenie údajov v MySQL databáze

Tabuľky v databáze

Dôležitou súčasťou aplikácie je ukladanie záznamov. Na ukladanie záznamov je použitá MySQL databáza (odsek 2.2), ktorá bude uchovávať potrebné informácie o užívateľoch a potrebných dátach.

Do marketingovej aplikácie je možné sa prihlásiť, takže je nutné uchovávať prihlasovacie údaje užívateľov. Na to slúži tabuľka s názvom bakalarkaUserInfo. Tabuľka je vytvorená nasledujúcim SQL príkazom:

```
1 CREATE TABLE `bakalarkaUserInfo` (  
2   `id` int( 11 ) NOT NULL AUTO_INCREMENT ,  
3   `email` varchar( 30 ) NOT NULL ,  
4   `heslo` varchar( 30 ) NOT NULL ,  
5   `token` text NOT NULL ,  
6   `entrance` tinyint( 1 ) NOT NULL DEFAULT '0',  
7   `inside` tinyint( 1 ) NOT NULL DEFAULT '0',  
8   `checkout` tinyint( 1 ) NOT NULL DEFAULT '0',  
9   PRIMARY KEY ( `id` )  
10 ) ENGINE = InnoDB DEFAULT CHARSET = utf8;
```

Obr. 3.8: SQL príkaz na tvorbu tabuľky užívateľov

Týmto príkazom je vytvorená tabuľka obsahujúca premenné id, email, heslo, token, entrance, inside, checkout. Viac informácií o jednotlivých premenných v tabuľke užívateľov je možné získať v Tab. 3.1

Tabuľka užívateľov obsahuje údaje o užívateľovi. Údaje ako je meno, hash z hesla, token slúžiaci k odoslaniu notifikácie priamo na jedno zariadenie, údaje o pohybe užívateľa a unikátne id užívateľa. Údaje o pohybe užívateľa sa skladajú z troch premenných typu boolean:

Tab. 3.1: Bližšie informácie k tabuľke užívateľov

	Typ	Dôvod	Funkcia
id	max 11 ciferne číslo	neobmedzený počet používateľov	unikátny indentifikátor užívateľa id sa automaticky inkrementuje o 1
email	max 30 znakov	obmedzené na 30 znakov pre šetrenie miesta v databáze	prihlasovací email užívateľa
heslo	max 256 znakov	hash z hesla užívateľa	zahashované heslo užívateľa z dôvodu bezpečnosti
token	neobmedzený text	unikátny indentifikátor zariadenia	adresa pre push notifikáciu na zariadenie
entrance	boolean	možnosti len 0 a 1 presne ako boolean	informácia o tom, či užívateľ prišiel k predajni
inside	boolean	možnosti len 0 a 1 presne ako boolean	informácia o tom, či užívateľ vošiel do predajne
checkout	boolean	možnosti len 0 a 1 presne ako boolean	informácia o tom, či užívateľ zaplatil

- premenná `entrance`

zmení sa na `true`, keď sa užívateľ priblíži k BEACONU1

- premenná `inside`

zmení sa na `true`, keď sa užívateľ priblíži k BEACONU2

- `checkout`

zmení sa na `true`, keď sa užívateľ priblíži k BEACONU3

Hash je funkcia ktorá s akéhokoľvek reťazca vygeneruje string, ktorý má vždy 256 znakov. Jedná sa o sekvenciu znakov z ktorej sa útočník nevie dostať k prvej zadanému heslu užívateľa. V bakalárskej práci sa používa hashovacia funkcia SHA-1. Kvôli univerzálnosti využitia beacnov, hlavne aby bolo možné konfigurovať beacons na väčšie vzdialenosti, bolo nutné dôležité informácie preniesť z hardwaru beacnu do databáz. Tým pádom je treba vytvoriť miesto na uloženie dát do databázy. Tieto informácie sa ukládajú v MySQL tabuľke s názvom `bakalarkaTags`, ktorú je možné vytvoriť zadaním nasledujúceho SQL príkazu na obrázku 3.9:

Bližšie informácie o premenných v tabuľke sú vypísané v Tab. 3.2. Na odosielať notifikácií musí byť zriadená ďalšia tabuľka s informáciami ktoré sa zobrazia v notifikácií, aká zľava, na aký produkt, koľko stojí a podobne. Doprovodnou tabuľkou je tabuľka s informáciami, komu a kedy prišla notifikácia, poprípade kedy platnosť vyprší. Jedná sa o doporovdné tabuľky takže netreba uvádzať ich štruktúru. Ich štruktúra závisí od potrieb a zamerania aplikácie.

Táto tabuľka obsahuje všetky informácie priradené k beacnu. Táto tabuľka vznikla ako náhrada za využitie Kontakt.io cloudu, ktoré obsluhovalo tagy priradené k be-

```

1 CREATE TABLE `d157971_gallop`.`bakalarkaTags2` (
2   `id` int( 11 ) unsigned NOT NULL AUTO_INCREMENT ,
3   `beaconUniqueId` varchar( 5 ) NOT NULL ,
4   `tagId` int( 11 ) NOT NULL ,
5   `tagName` varchar( 20 ) DEFAULT NULL ,
6   `tagMinRSSI` int( 11 ) NOT NULL ,
7   PRIMARY KEY ( `id` ) ,
8   UNIQUE KEY `id` ( `id` ) ,
9   KEY `id_2` ( `id` )
10 ) ENGINE = InnoDB DEFAULT CHARSET = utf8;
11

```

Obr. 3.9: SQL príkaz na tvorbu tabuľky tagov

Tab. 3.2: Bližšie informácie k tabuľke tagov

	Typ	Dôvod	Funkcia
id	max 11 ciferne číslo	neobmedzený počet beacnov a tagov k nemu	unikátny indentifikátor v tabuľke id sa automaticky inkrementuje o 1
beaconUniqueId	max 5 znakov	obmedzené na 5 znakov pre šetrenie miesta v databáze	unikátne id beacnu (určené hardware-om)
tagId	max 11 ciferne číslo	neobmedzený počet tagov k beacnu	id tagu v beacne, poradie dôležité pre chod aplikácie
tagName	max 20	meno endpointu	endpoint ktorý bude kontaktovaný po priblížení k beacnu
tagMinRSSI	max 11 ciferne číslo	akékoľvek číslo	RSSI hodnota ktorú musí beacon spĺňať aby prebehla komunikácia so serverom

acnom. Kontakt.io mal v čase vývoja aplikácie problémy v ich zverejnenom API na konfiguráciu tagov k beacnu, tak bolo treba premiestniť informácie o tagoch z ich správy do našej. V konečnom dôsledku to prináša množstvo výhod. Jednou z nich je možnosť využitia jedného beacnu vo viacerých aplikáciách s inou funkčnosťou. Ďalšou výhodou je možnosť nastavenia či je nutné stiahnuť všetky existujúce tagy alebo len všetky tagy k príslušnej aplikácii, či dokonca len tagy k príslušnej aplikácii a beacnu v okolí. V prípade uloženia tagov vo vlastnej databáze je možnosť uloženia viac informácií než len tag napr už spomínané major, minor id, url alebo akékoľvek ďalšie informácie podľa potreby a to je možné dynamicky meniť za behu aplikácie bez nutného kontaktu s beacnom (bez nutnosti konfigurovať beacon cez Bluetooth). Stahovanie a spracovanie prijatého tagu je z vlastnej databázy citeľne rýchlejšie pokiaľ sa jedná o menší počet tagov avšak dosť záleží na výkonnosti zariadenia.

Ďalšia doprovodná tabuľka zhromažďuje všetky zľavové ponuky, ktoré je možné užívateľovi odoslať. Obsahuje id zľavy, názov kupónu, opis kupónu, cestu k obrázku, zjednodušený opis kupónu a hodnotu kupónu. Z tejto tabuľky sa berie zľavová ponuka, ktorá bude užívateľovi doručená, či už na základe informácií z pohybu užívateľa

alebo náhodného výberu.

Doprovodná dohľadová tabuľka zhromažďuje históriu odoslaných kupónov všetkých užívateľov. Obsahuje unikátne id notifikácie, id užívateľa ktorému bola notifikácia odoslaná, id zľavovej ponuky a čas začiatku a konca platnosti daného kupónu.

Konfigurácia

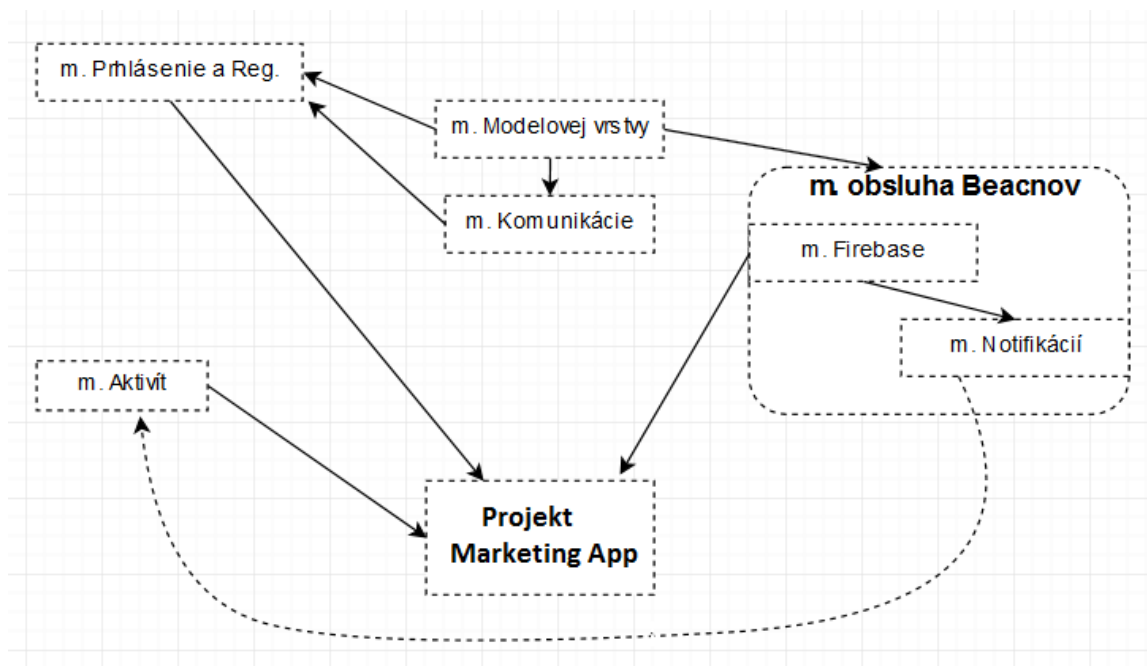
Každý beacon musí byť zaradený najmenej do jedného tagu, tagy musia obsahovať slovo, ktoré bude niesť informáciu na aký endpoint je treba odoslať žiadosť na server a limitnú hodnotu prijatého výkonu z beaconu. Meraním ktoré bolo uskutočnené v rámci bakalárskej práce bolo zistené, že nie je možné sa spoliehať na dosah určený nastavením levelu výkonu, preto sa do tagu vpisuje hodnota, ktorá značí minimálny možný prijatý výkon z beacnu na to, aby bolo umožnené odoslať žiadosť na server. Zaradenie beacnu do tagu, z ktorého sa berú dáta potrebné na ďalšiu komunikáciu je najlepším riešením na správu beacnov, keďže tagy nie je potrebné zapisovať cez bluetooth do zariadenia. Zmeniť komunikačné endpointy a limitné údaje na overenie vzdialenosti od beacnu bude možné pohodlne na diaľku z webového rozhrania alebo priamym prepisom hodnôt v tabuľke tagov.

3.2.3 Rozdelenie Android aplikácie na časti - moduly

Marketingová aplikácia sa skladá z viacerých modulov. Modul je samotný projekt na riešenie čiastočného problému. Prehľad o moduloch v projekte Marketingovej aplikácie aj so závislosťami je k nahliadnutiu na obr. 3.10.

Prihlásenie a registrácia

Modul sa skladá z obrazovky na prihlásenie a registráciu. Komunikácia daného modulu s databázou je na Obr. 3.13. Jedná sa o aktivitu ktorá obsahuje jeden z dvoch fragmentov. Implicitne sa jedná o aktivitu registrácie 3.11, v ktorej sa užívateľ vyplnením emailu a dvojstupňovým vyplnením hesla zaregistruje alebo sa prepne na obrazovku, kde sa prihlási viz Obr. 3.12. Prepnutím obrazovky sa zmení fragment, z fragmentu na registráciu na prihlasovací fragment. Prvým bodom komunikácie je vyplnenie emailu a hesla užívateľom. Po úspešnom zvalidovaní vyplnených údajov (email vo formáte emailu, pri registrácii zhodovanie hesiel). V druhom bode sa zahasuje heslo užívateľa a spolu s emailom sa odošle žiadosť na databázu a komunikáciu so serverom. Komunikácia s databázou na strane smartfónu je vykonávaná nasledujúcim kódom 3.2. Do parametrov funkcie sa vkladá URL adresa, modelová trieda užívateľa (obsahujúca meno, priezvisko, heslo a token), handler, ktorý oznámi aplikácii úspešnosť/neúspešnosť daného requestu a context k aktivite ktorá vyvoláva



Obr. 3.10: Znázornenie závislostí modulov v projekte

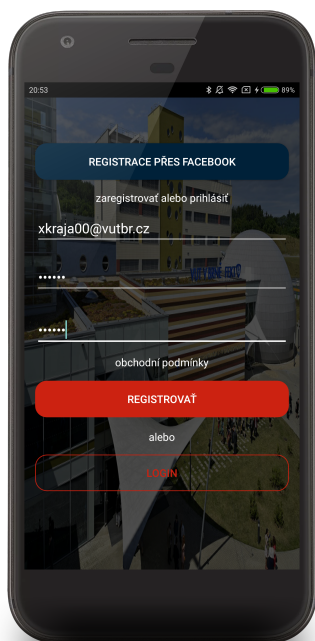
komunikáciu. Je nadväzované spojenie s url uvedenou v parametre funkcie, ďalej sa kontaktuje daná url a vkladajú sa do nej dáta, ktoré sú uložené v premennej `nameValuePairs`. Je to štruktúra, ktorá sa pri odosielaní automaticky prepíše do tvaru JSONu 2.4 čiže dvojíc meno a hodnota.

Tretím bodom komunikácie je kontrola, či je užívateľ registrovaný, následne či vyplnil správne prihlasovacie údaje. V každom prípade príde späť do smartfónu odovď, či už vo forme „objektu“ prihláseného užívateľa alebo nejakého z chybových kódov. Odpoveď zo servera vložená do premennej `responseStr`, ktorá je vložená do handleru, ktorý je v inej triede odchytý a výsledok zobrazený užívateľovi. V prípade chybového kódu je užívateľovi oznámené o akú chybu sa jedná (užívateľ neexistuje alebo zlé heslo) v prípade úspešného prihlásenia je užívateľ automaticky presmerovaný do aplikácie.

```

override fun onEddystoneDiscovered(eddystone: IEddystoneDevice?,
    namespace: IEddystoneNamespace?) {...}
override fun onEddystonesUpdated(eddystones: List<IEddystoneDevice>?,
    namespace: IEddystoneNamespace?) {...}
override fun onEddystoneLost(eddystone: IEddystoneDevice?, namespace:
    IEddystoneNamespace?) {...}
BeaconHelpers.Companion.fetchTags(device.uniqueId,this@MainActivity,(
BeaconHelpers.Companion.workWithDataFromBeacon(this@MainActivity,

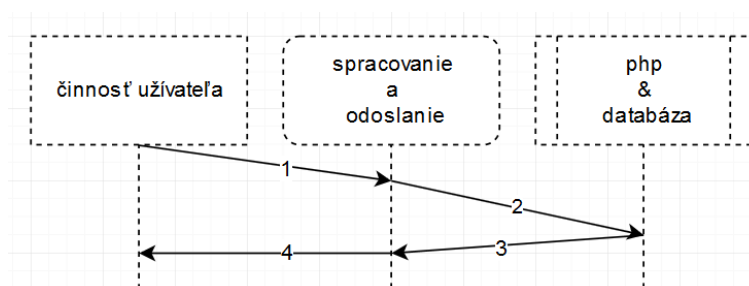
```



Obr. 3.11: Registrácia nového užívateľa do aplikácie



Obr. 3.12: Prihlásenie užívateľa do aplikácie



Obr. 3.13: Postup komunikácie prihlásenia

```
"NOTIFICATION_BEACON_CONFIG", device)))
```

Výpis 3.1: Tvorba listeneru pre Eddystone beacony

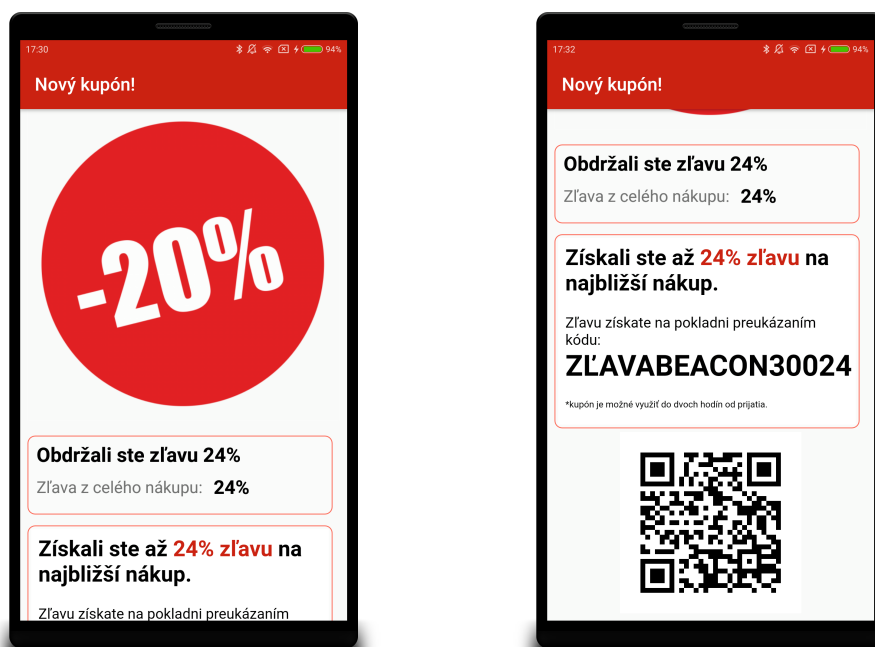
Modelová vrstva

V tejto časti je opísaný modul modelovej vrstvy. Tento modul obsahuje modelové triedy používané v aplikácii. Modelové triedy slúžia na vytvorenie objektu z JSONu, ktorý je prijatý zo servera. Pomocou modelových tried sú lokálne uložené tagy, ktoré sú v modelovej triede beaconu, ktorá je uložená v Realm. Rovnako pomocou modelových tried sú v zariadení uložené informácie o prihlásenom užívateľovi. Modelová

vrstva neobsahuje žiadnu komplikovanú logiku, je to len trieda so správne pomenovanými premennými -> premenné musia mať rovnaký názov v modelovej triede ako v prijatom JSONe a ak už sa nevolajú rovnako treba premennej priradiť jej príslušnú anotáciu na deserializér.

Modul aktivít

V tomto module sú uložené aktivity používané v aplikácii. Dôležitou aktivitou v tomto module je aktivita detailu obdržaného kupónu z notifikácie 3.14. Je to aktivita ktorá sa otvorí po kliknutí na notifikáciu zobrazenú v lište notifikácií. Aktivita implementuje generovanie QR kódov na jednoduché naskenovanie pri použití zľavových kupónov. Ďalšou aktivitou je aktivita, ktorá zobrazuje obdržané kupóny. Ďalšou nemenej dôležitou aktivitou v aplikácii je aktivita 3.15, ktorá zobrazuje všetky obdržané kupóny s možnosťou prekliknutia na detail 3.14.

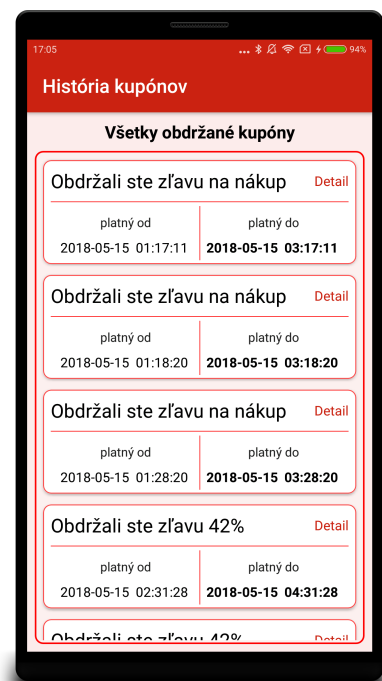


Obr. 3.14: Detail obdržanej notifikácie

Komunikačný modul

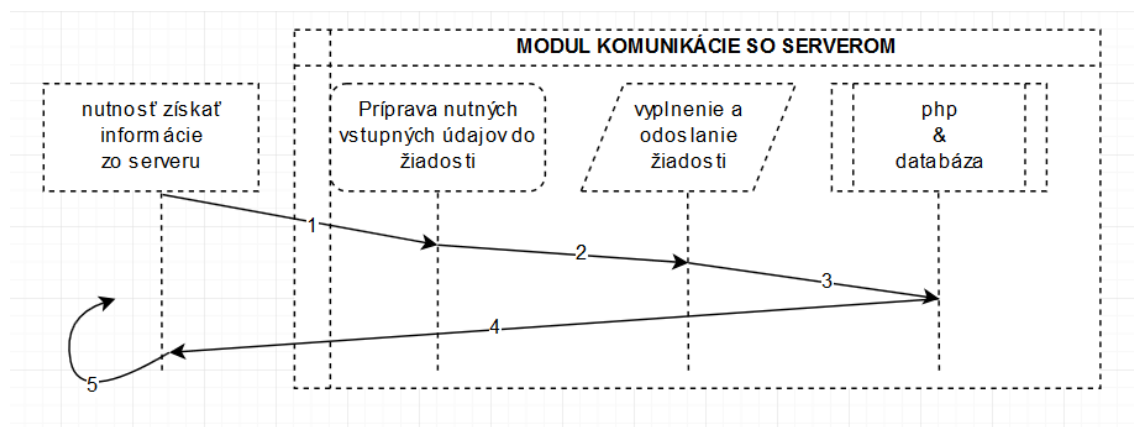
V tomto module sa nachádzajú univerzálne funkcie a metódy na komunikáciu so serverom. Implementácia modulu do aplikácie je znázornená na diagrame zobrazujúcom komunikáciu so serverom 3.16. Keď je nutné dostať dáta zo serveru je potrebné využiť komunikačný modul.

1. Zhromaždenie údajov potrebných pre identifikáciu žiadosti a vyplnenie údajov. (Ktorý endpoint je nutné kontaktovať a dáta ktoré je nutné odoslať v ňom.)



Obr. 3.15: Aktivita zobrazujúca všetky obdržané kupóny

2. Spracovanie dát prijatých do modulu, vloženie dát nutných pre endpoint a kontaktovanie požadovaného endpointu.
3. Odoslanie žiadosti na server.
4. Server odošle žiadateľovi požadované dáta alebo chybový kód
5. V prípade nutnosti je odpoveď zo servera odoslaná do ďalšieho modulu na spracovanie.



Obr. 3.16: Znázornenie komunikácie v sekvenčnom diagrame

Modul na obsluhu Beacnov

Všetky úkony okolo beacnov sú uložené v jednom module, ktorý obsahuje dva sub-moduly a jednu inicializačnú triedu. Inicializačná trieda v sebe obsahuje všetky komponenty, ktoré je nutné inicializovať pre prácu s týmto modulom. Tento modul je detailnejšie vysvetlený neskôr.

Firebase modul - modul implementujúci Firebase notifikácie. Obsahuje servis generujúci unikátny token pre doručenie notifikácie priamo na zariadenie, z ktorého je generovaný. Ďalej obsahuje servis sledujúci doručenie notifikácie. Ak je odoslaná notifikácia priamo na unikátny token pomocou tohto servisu je možné túto notifikáciu aj dáta odchytiť a pripraviť na spracovanie v ďalšej aplikácii.

Modul notifikácií - modul zastrešujúci zobrazenie notifikácií a konfigurácia udalostí, ktoré nastanú po kliknutí alebo zmazaní notifikácie.

Projekt Marketing app

Projekt marketing app importuje moduly ktoré používa. Pomocou jednoduchšej inicializačnej konfigurácie prepája moduly tak, ako je v projekte vyžadované.

3.2.4 Autentizácia

Autenticácia užívateľa prebieha využitím dvoch modulov, jedná sa o modul prihlásenia a registrácie a modelu komunikácie. Z modulu prihlásenia a registrácie je využívané UI, prepojenie prvkov a kontrola validnosti vyplnených dát a z modulu komunikácie sa kontaktuje server s prihlasovacími údajmi. Táto komunikácia prebieha cez zabezpečený kanál https. Jedná sa o zabezpečenú komunikáciu, ktorá používa platný bezpečnostný certifikát. Je možné implementovať zabezpečenie pomocou časového odtlačku, ktorý je možné pridať za každé odoslanie prihlasovacích údajov a na strane kontrol, či sa užívateľ so zadanými údajmi nachádza v databáze aby sa najskôr overila platnosť prijatých dát. Overenie by prebehlo spôsobom či časový odtlačok by sa nachádzal v tolerancií. Je nutné počítať s časom uplynutým od zashovania časového odtlačku po kontrolu odtlačku. Odoslanie a prijatie môže trvať od pár milisekúnd po desiatiny sekundy, záleží od toho aké veľké dáta sa odosiľajú.

```
public void doFileUpload(final String url, final RealmModelUser user,
    final Handler handler, Context context) {\n
    String meno = user.getMeno();
    ...
}
```

```

final ArrayList<NameValuePair> nameValuePairs = new
    ArrayList<NameValuePair>();
nameValuePairs.add(new BasicNameValuePair("meno", meno));
...
httppost.setEntity(new UrlEncodedFormEntity(nameValuePairs));
HttpResponse response = httpclient.execute(httppost);
String responseStr = EntityUtils.toString(response.getEntity());
...
bundle.putString("response", responseStr);
msg.setData(bundle);
handler.sendMessage(msg);\}

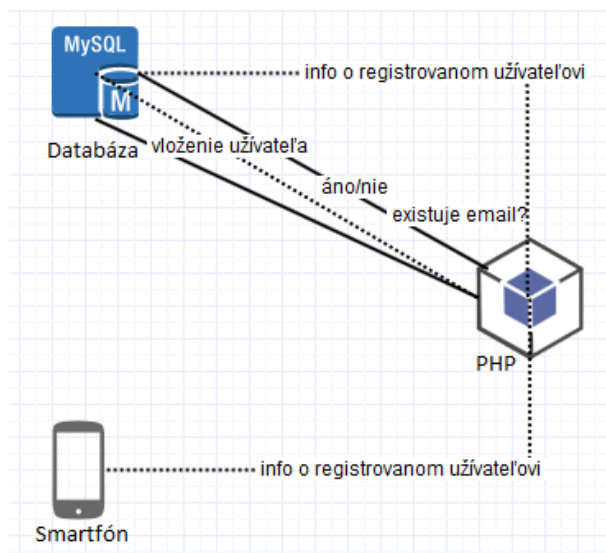
```

Výpis 3.2: Odosielanie prihlasovacích údajov na server

Komunikácia pokračuje ďalej, je nutné uložiť odoslané informácie do databázy. To prebehne pomocou skriptu v PHP, ktorého účel je zobrazený na Obr. 3.17. Skript funguje následovne: najskôr sa do premenných \$meno, \$email, \$token sa vložia potrebné dáta. Najskôr sa vyhladá email v databáze (zistenie či sa neregistruje registrovaný užívateľ) ak sa tam email ešte nenachádza, tak sa tam vloží riadok s užívateľovým menom, emailom, hashom z hesla a tokenom. Užívateľ je zaregistrovaný, v response je novoregistrovaný užívateľ. V prípade existujúceho emailu v databáze odpoveď obsahuje číslo 2, v prípade chyby pri vkladaní riadku do databáze je tam číslo 1.

Novo registrovaný užívateľ musí svoju emailovú adresu potvrdiť na zadanom emaily. Na email mu príde email s linkom, na ktorý musí kliknúť aby bola registrácia dokončená. Po overení emailu je možné sa bez problémov prihlásiť do aplikácie. V prípade, že užívateľ ešte nedokázal, že email s ktorým sa pokúša zaregistrovať mu nepatrí, tak po odoslaní žiadosti na prihlásenie sa nahliadne do tabuľky užívateľov do stĺpca authorized, ktorý indikuje stav, či je užívateľ overený alebo nie. V prípade že sa tam nenachádza 0, užívateľovi bude oznámené, že ešte neoveril email a nedostane sa do aplikácie. Táto kontrola emailu je užitočná pre používateľov, ktorý nechcú mať s aplikáciou nič spoločné. Keby bol užívateľ ohľadom zľavových ponúk alebo nejakým iným veciam kontaktovaný pomocou emailu, aby nebol kontktovaný užívateľ, ktorý nevie, že email aplikácii poskytol. Rovnako je to prevencia pred situáciou, že užívateľ sa chce zaregistrovať do aplikácie a vypíše mu, že: „Tento email sa už používa“ aj napriek tomu, že sa s aplikáciou stretol prvý krát.

V prípade, že užívateľ je úspešne registrovaný, v odpovedi requestu sú údaje o užívateľovi (za predpokladu, že overil email), ktoré sa opäť pomocou JSONu dostanú do telefónu, z JSONu sa vyexportujú potrebné údaje a uložia sa do Realmu.



Obr. 3.17: PHP skript registrácia

Po úspešnej transakcii ukladania do Realmu je užívateľ presmerovaný na ďalšiu obrazovku, kde je už aktívne detegovanie beacnov. 3.15

V prípade prihlasovania užívateľa sa kontaktuje podobný PHP skript, ktorý kontroluje len či zadaný email v tabuľke súhlasí so zadaným hashom v tabuľke. Ak áno v response sú informácie o userovi, ak nie vráti sa 1, čo značí chyba pri prihlasovaní (zlé meno alebo heslo).

3.2.5 Spracovanie nájdeného beacnu

Najdôležitejšie je vedieť spracovať informácie, ktoré sú v nájdenom beacne. V tejto kapitole je podrobnejšie opísaná štruktúra, triedy, metódy a funkcie ktoré obsluhujú aplikáciu. Spracovanie dát z beacnu je rozdelené do viacerých metód, ktorými je spracovanie rozdelené na viacero častí. Tu budú opísané len dôležitejšie funkcie na chod aplikácie.

Keď je užívateľ úspešne prihlásený a je na obrazovke, ktorá detekuje beacny, tak prvá vec, ktorú je nutné spraviť je ziniclizovať Kontakt.io SDK (je nutné mať Kontakt.io key). To sa robí jednoduchým príkazom na inicializáciu. Ďalej je nutné vyžiadať od užívateľa povolenie na prístup k polohe zariadenia (vyžaduje si to Kontakt SDK) a povolenie k prístupu k Bluetoothu. Ak je prístup povolený ziniclizuje sa vytvorený singleton na detekciu beaconov v okolí. Singleton je trieda, ktorá sa inicializuje raz a potom je možné sa dostať k rovnakej inštancii po celý beh aplikácie. V tomto singletone je uložený listener, ktorý si užívateľ zvolil, keď ho inicializoval. Zobrazovať singleton nie je nutné, hlavné je že vstupným parametrom pri jeho volaní je listener na iBeacon alebo Eddystone beacon, poprípade na oba.

Na tvorbu listeneru je vytvorená metóda ktorá deteguje tri možné stavy beacnov ktoré vysielajú Eddystone protokol, `onEddystoneDiscovered`, `onEddystoneUpdated`, `onEddystoneLost`. `onEddystoneDiscovered` sa vyvolá ak smartfón nájde v dosahu nový `EddystoneBeacon`, dáta v nájdenom beacne sú reprezentované v premennej `eddystone`. Z tejto premennej je nutné dostať `uniqueId` beacnu a to pomocou príkazu `eddystone?.uniqueId`. V Obr. 3.1 dole je použitá funkcia vo vytvorenej triede `BeaconHelpers`, čo je doprovoďná zastrešujúca stiahnutie tagov z MySQL databáze. Ďalšími vstupnými parametrami sú kontext aktivity, z ktorej je metóda volaná. Posledný vstupný parameter je callback, to je metóda, ktorá sa zavolá ak sú stiahnuté tagy k nájdenému beacnu. Je možné spracovávať beacons, ktoré majú `uniqueId`, nie všetky ho majú, napríklad beacons od výrobcu proxiot `uniqueId` nemá, ale má MAC adresy (aj beacons od Kontaktu majú MAC adresu, ale je zbytočne dlhá na spracovávanie a ukladanie v databáze, navyše `uniqueId` sa ľahko pamätá a lepšie sa vyzná v beacnoch).

- ***OnEddystoneUpdate***

sa volá po opätovnom "nájdení" beacnu/beacnov, je možné s nimi znovu pracovať. V databáze sa mohli zmeniť podmienky práce s beacnom a aplikácia môže reagovať na opätovné nájdenie inak než na prvotné nájdenie.

- ***OnEddystoneDiscovered***

sa môže volať neobmedzený počet krát pokiaľ sa nespustí časť `OnBeaconLost`, ktorá informuje, že beacon sa stratil z dosahu. Potom je nutné najskôr `onEddystoneDiscovered` a následne sa môže znovu volať `onEddystoneUpdated`.

- ***OnEddystoneUpdated***

a `onEddystoneDiscovered` je v podstate to isté volanie, len napríklad v použití v aplikácii na marketing je možné v `onEddystoneDiscovered` stiahnuť všetky tagy k beacnu a v `onEddystoneUpdate` budú VŽDY stiahnuté. Aj to tak je, len overovanie či sú tagy stiahnuté prebieha vo vnútri metód a funkcií.

Ďalšou dôležitou funkciou je `checkToleranceDistance` 3.3. Jedná sa o funkciu na teoretický výpočet vzdialenosti od beacnu, jej vstupné parametre sú kontext aktivity, z ktorej je funkcia volaná, trieda typu `RemoteBluetoothDevice`, `configKey`, `endpoint`, s ktorým bude telefón komunikovať a minimálna hodnota rssi na spustenie komunikácie medzi telefónom a serverom. V prvom riadku funkcie sa zisťuje `Id` prihláseného užívateľa, najskôr sa v Realm 2.6 (lokálnej databáze) pomocou query vyhladá prihlásený užívateľ, následne sa pomocou `.id` zistí jeho `id`, ktoré sa uloží do premennej `id`. Nasleduje podmienka, ktorá je pravdivá v momente keď hodnota rssi nájdeného beacnu je väčšia než hodnota udaná v premennej `valueTag`. 3.2 Ak je táto podmienka splnená zavolá metóda `sendUserRoute` 3.4. Táto metóda kontaktuje

server na endpoint, ktorý je uvedený v tagu, tag neobsahuje celú adresu k tagu ale len špeciálne ukončenie, lebo základná cesta je pre všetky endpointy rovnaká. Tak namiesto dlhého odkazu je v ňom ukladané len pár posledných znakov, ktoré sa spolu spoja v druhom riadku funkcie. Následne sa kontaktuje server pomocou pripravenej asynchronnej komunikácie podobnej tej na prihlásenie a registráciu užívateľa s tým rozdielom, že sa odosiela len id užívateľa.

```
fun checkToleranceDistance(context: Context, device:
    RemoteBluetoothDevice, configKey: String, typeTag: String?, valueTag:
    Int) { val id =
        Realm.getDefaultInstance().where(RealmModelActiveUser::class.java)
        .findFirst()!!.id
    if (valueTag <= device.rssi) {sendUserRoute(context, configKey,
        typeTag!!, id)} }
```

Výpis 3.3: Funkcia na teoretický výpočet vzdialenosti od beacnu

```
fun sendUserRoute(context: Context, configKey: String, tagadress: String,
    id: Int) {
    val notificationbeaconConfig =
        ConfigurableManager.getInstance().getConfigurable(configKey) as
        NotificationBeaconConfig
    val dataPair = Pair(String.format("\s/%s.php",
        notificationbeaconConfig.prefixUrl, tagadress), id)
    object : ServerBackgroundGetUserWay(context) {}.execute(dataPair)}
```

Výpis 3.4: Funkcia zisťujúca smer, v ktorom užívateľ prechádza okolo beacnu

Obe tieto funkcie sú použité vo funkcii `workWithDataFromBeacon` 3.5, ktorá sa spustí po úspešnom stiahnutí a uložení tagov 3.1. Táto funkcia sa nachádza v module na obsluhu beacnov. Najskôr je skontrolované, či `uniqueId` nájdeného Bluetooth zariadenia nie je `null` (je možné nájsť iný telefón, slúchadlá, televízor čokoľvek čo je v okolí a má zapnutý Bluetooth), ak sa `uniqueId` nerovná `null`, tak sa toto id znovu pomocou query nájde v uložených beacnoch. Ak sa nájde znamená to, že určite je k nemu priradený aspoň jeden tag. (v prípade, že sa nenájde znamená to že je to Bluetooth zariadenie, ktoré nie je využívané v aplikácii, čiže sa nenachádza v tabuľke `bakalarkaTags` 3.9) Ak je to beacon patriaci k aplikácii, tak sa vykoná ďalší príkaz, ním sa znovu v Realm nájde užívateľ a zistí jeho id. V premennej objektu `found` je premenná `tags`, čo je list s tagmi, ktoré sú priradené k danému beacnu v databáze. Ako je možné vidieť, najskôr sa berie tag, ktorý je v liste s najväčším indexom. Čiže v premennej `valueTag` a `typeTag` sú informácie tagu s najvyšším `tagId` 3.2

Ďalej sa znovu kontroluje, či aktuálne nájdený beacon má vyhovujúcu hodnotu

RSSI, ak áno tak sa znovu naplnia informácie (endpoint a id) potrebné k začiatku komunikácie medzi smartfónom a serverom a spustí sa univerzálna metóda na komunikáciu so serverom, kde sa do requestu vkladá len id užívateľa. V odpovedi je väčšinou nejaké číslo, ktoré hovorí o stave vykonania požiadavky, chyba, úspech alebo nejaký objekt v JSONe. Napríklad v prípade beacnu nad vchodom/východom z predajne je z tagu s najvyšším idTag odpoveďou buď číslo 0, ktoré znamená, že užívateľ ide do predajne alebo 1, ktoré znamená že užívateľ z predajne vychádza. V skratke sa číta hodnota `inside` z tabuľky `bakalarkaUserInfo` a to pomocou php skriptu ktorý sa kontaktuje práve pomocou tagu.

Neskôr ak je v response číslo, ktoré je menšie než najmenšia zo stavových konštánt (200) tak to znamená, že sa jedná o pozíciu tagu ktorú treba následne kontaktovať. Takže, v prípade, že užívateľ prichádza do predajne opätovne sa kontaktuje tag s rovnakého beacnu avšak tag s indexom 0 a nastaví sa hodnota `entrance` v databáze na 1. V prípade, že užívateľ odchádza, tak sa kontaktuje tag s id 1, čo znamená že sa treba pozrieť do databázy na hodnotu `checkout` ktorá ak je 0 (užívateľ nešiel okolo pokladní) tak sa automaticky pomocou skriptu 3.18 tvorí notifikácia, ktorá je doručená užívateľovi priamo na telefón. Po ukončení odosielania notifikácie je v response číslo 200, rovnako aj v prípade že užívateľ šiel okolo pokladní len s tým rozdielom, že nie je vygenerovaná notifikácia, ale sú všetky premenné (`entrance`, `inside`, `checkout`) nastavené na 0, čím sú pripravené na ďalšiu návštevu užívateľa. Ak sa užívateľ znovu vráti do predajne celý kolobeh sa opakuje. V prípade obmedzenia zľavových kupónov na smartfón na jeden deň je možné jednoducho do tabuľky užívateľa vložiť stĺpec, ktorý bude indikovať kedy bol užívateľovi doručený kupón naposledy. (Alebo je nutné prejsť tabuľku so všetkými odoslanými kupónmi. Tam sa ukladá kto dostal, kedy a aký kupón.)

```
if (bleDevice.uniqueId != null) {
    val found=Realm.getDefaultInstance().where(MyDevice::class.java)
        .equalTo("uniqueID", bleDevice.uniqueId).findFirst()
    if (found!=null) {
        val
            id=Realm.getDefaultInstance().where(RealmModelActiveUser::class.java)
                .findFirst()!!.id
        val typeTag=found.tags!![found.tags!!.size - 1]!!.tagName
        val valueTag=-1*found.tags!![found.tags!!.size - 1]!!.tagMinRSSI
        if (valueTag<=bleDevice.rssi) {
            val notificationbeaconConfig=ConfigurableManager.getInstance()
                .getConfigurable(configKey) as NotificationBeaconConfig
            val dataPair=Pair(String.format("%s/%s.php",
                notificationbeaconConfig.prefixUrl, typeTag), id)
```

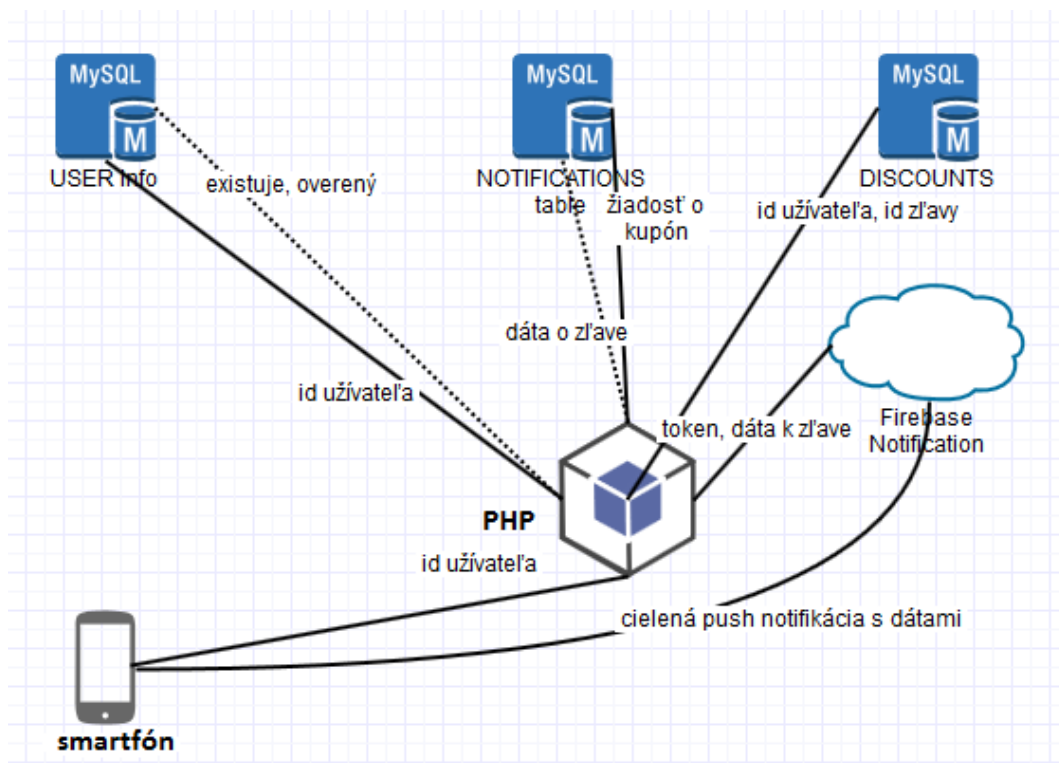
```

object : ServerBackgroundGetUserWay(context) {
    override fun onPostExecute(result: String) {
        val resInt=Integer.parseInt(result)
        if (resInt<CONST_OK_FROMSERVER)
            BeaconHelpers.Companion.checkToleranceDistance(context,
                bleDevice, configKey, found.tags!![resInt]!!.tagName,
                -1*found.tags!![resInt]!!.tagMinRSSI)}}.execute(dataPair)
    ...

```

Výpis 3.5: Funkcia spracujúca nájdený beacon

Schématický opis skriptu na vygenerovanie a odoslanie notifikácie Obr. 3.18, je pomerne obsiahnejší a skladá sa z množstva menších funkcií, ktoré implementuje. V prípade potreby vysvetlenia postupu tvorby notifikácie v php skripte prečítať [3].



Obr. 3.18: Funkcia skriptu tvoriaceho notifikáciu

3.2.6 Prijatie a spracovanie notifikácie

Po úspešnom vygenerovaní a odoslaní notifikácie sa pomocou platformy Firebase doručí do telefónu na základe tokenu uloženého v tabuľke užívateľov v databáze. Prijatá notifikácia sa odchyťava pomocou servisu. Ak servis zaregistruje prijatú správu odoslanú z Firebase zaregistruje ju listener `onMessageReceived`, najskôr treba

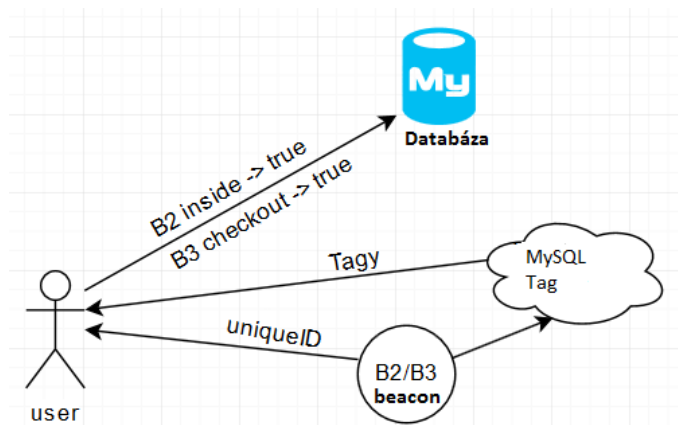
pozrieť, či prijatá správa obsahuje nejaké dáta. V prípade že áno tak sa z nich vytvorí JSONObject, ktorý je vložený do metódy kde sa notifikácia pripraví na zobrazenie užívateľovi.

Každá notifikácia používaná v aplikácii obsahuje premennú title, message, data a obrázok. Tieto informácie je možné získať ak je do projektu úspešne importovaný plugin od Firebase. Title je titulok zobrazený v notifikácii, message je menší text v zobrazenej notifikácii. V dátach sú informácie k produktu, ktorý je v zľave, image je v prípade že je prijatý veľký typ notifikácií, v ktorom je pod titulkom a správou veľký obrázok cez polovicu notifikačnej plochy. Informácie z prijatých dát sú uložené do lokalnej databázy, do Realmu a nastaví sa koniec platnosti kupónu na dve hodiny. Po úspešnom uložení sa vo vytvorenej triede notificationmanager zavolá metóda showSmallNotification alebo showBigNotification v závislosti od toho, či prijaté dáta obsahujú obrázok alebo nie. V týchto metódach sa vložia potrebné dáta do notifikácie ikona, zvuk pri prijatí, titulok, správa poprípade obrázok, dôležitým údajom je intent. Informácie v premennej intent obsahujú odkaz na aktivitu, ktorá sa otvorí po kliknutí na notifikáciu, v tomto prípade na detail prijatej zľavovej ponuky. V prípade Androidu väčšieho alebo rovného 8.0 je nutné vytvoriť pre tvorbu notifikácií špeciálny kanál.

3.3 Komunikácia v aplikácii

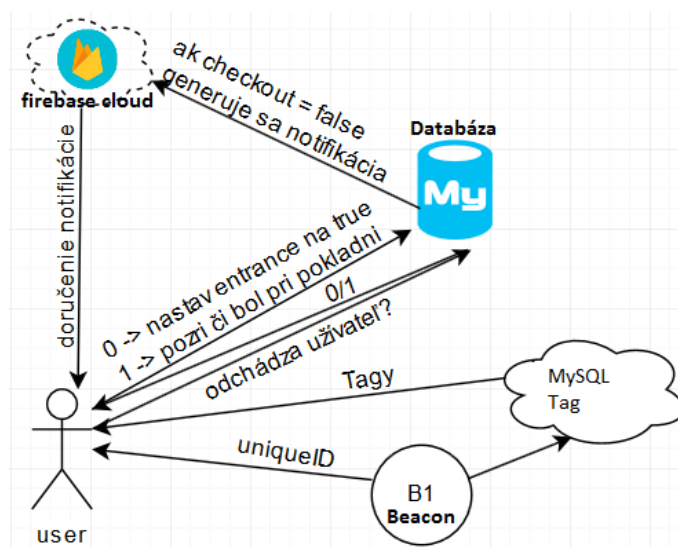
Aplikácia si bude držať a pravidelne aktualizovať tagy z okolitých beacnov aby nebolo treba vždy pri nájdení beacnu vyhľadať jeho tagy. Ďalej bude opísaná komunikácia medzi BEACONOM2, BEACONOM3 a serverom. Hneď ako sa smartfón dostane do dosahu daných beacnov, tak vyžiada si ich tagy, ktoré si rozdelí na endpoint a limitnú úroveň výkonu. Na záver si všetky tagy uloží do pamäte. Keď sa smartfón priblíži do dostatočnej blízkosti k beaconu tak sa na server odošle informácia o tom, že užívateľ s daným id sa priblížil k BEACONU2 alebo BEACONU3. V závislosti od toho, ktorému beacnu sa užívateľ priblížil, odošle žiadosť na server a to z dôvodu rozličných endpointoch v tagu beacnu. V prípade priblíženia sa k BEACONU2 sa odošle na server správa, ktorá v databáze v tabuľke užívateľov zmení hodnotu inside na true. Ak sa užívateľ priblíži k BEACONU3 tak sa v rovnakej tabuľke v rovnakom riadku zmení hodnota checkout na true. Náčrt komunikácie je schématicky vyobrazený na obr. 3.19.

Komplikovanejšia logika je implementovaná k beacnu s pracovným označením BEACON1, tento beacon patrí do troch tagov. Jeden z nich obsahuje presmerovanie na logický endpoint, ktorý vráti hodnotu v stĺpci inside (0 alebo 1) na základe odpovede sa odošle na server žiadosť na priradenie hodnoty true do stĺpca entrance ak



Obr. 3.19: Komunikácia medzi beacnami BEACON2 a BEACON3 so serverom

sa hodnota v stĺpci inside rovná false, ak sa hodnota v stĺpci inside rovná true, znamená to že užívateľ je na odchode z predajne. Ak je užívateľ na odchode z predajne, tak sa odošle žiadosť na endpoint, ktorý skontroluje, či hodnota v stĺpci checkout je true. Ak nie je znamená to, že užívateľ nešiel cez pokladne, takže sa začne proces generovania notifikácie. Najskôr treba zistiť token, vďaka ktorému sa notifikácia doručí priamo na zariadenie užívateľa, potom treba vybrať id zlavového kupónu, ktorý bude doručený užívateľovi. Script odošle tieto údaje do Firebase cloudu, ktorý doručí dané informácie priamo k užívateľovi, ktorý správu prijme v podobe JSONu z ktorého sa dáta uložia do modelovej triedy v Realm. V ňom budú uložené. Keď prebehne uloženie, užívateľovi bude zobrazená notifikácia v hornej lište smartfónu. Keď užívateľ klikne na notifikáciu bude presmerovaný na detail zlavovej ponuky, ktorú práve obdržal. Táto komunikácia je načrtnutá na obr. 3.20



Obr. 3.20: Komunikácia medzi beacnom BEACON1 a serverom

4 ZÁVER

V úvode práce bol rozobraný súčasný stav identifikačných technológií a analýza ich vlastností. Na základe tejto analýzy sa javí Bluetooth Low Energy a Beacon ako najperspektívnejšia technológia pre marketingové účely.

V ďalšej časti bola navrhnutá a implementovaná aplikácia pre meranie prijatého výkonu a chybovosti spracovania prijatého signálu v závislosti na vzdialenosti. Z meraní je na testovacích zariadeniach je jasné, že nie je vhodné spoliehať sa na určovanie vzdialenosti od beacnu na základe poznania výkonu a detegovateľnosti, je oveľa lepšie odhadovať vzdialenosť na základe úrovne rssi detegovaného beacnu.

Druhou aplikáciou bola aplikácia zaštitujúca cieľný marketing na užívateľa vzhľadom na jeho pohyb v obchode. Do aplikácie bolo implementované aj zistenie z merania o spoľahlivosti beacnov. To tak, že každému beacnu je priradený tag s informáciou na ktorý endpoint má odoslať žiadosť a minimálny prijatý výkon pri ktorom je možné danú žiadosť odoslať. Toto riešenie považujem za najlepšie, keďže tagy nie sú zapísané priamo v beacnoch, ale v mojej databáze, takže sa dajú meniť bez toho aby bola nutná konfigurácia beacnu cez bluetooth. Podrobnejšie informácie k vytvoreným aplikáciám sa nachádzajú v tretej kapitole 3.

Výsledkom bakalárskej práce je modul implementujúci univerzálnu logiku na správu beacnov. Nastavenie udalostí, ktoré majú nastať pri priblížení k beacnu je čisto na osobe ktorá tento modul implementuje. Správanie jednotlivých beacnov sa jednoducho nastavuje cez tagy, ktoré sú beacnom priradené. Keďže tagy v sebe obsahujú podmienku a komunikačný endpoint na serveri, tak celá logika aplikácie implementujúcej daný modul je na strane serveru. V prípade beacnu, ktorý môže mať trojaké správanie v závislosti od dát na serveri je nutné dodržať len poradie priradených tagov (posledný tag sa vykoná ako prvý, ďalšie v závislosti od odpovede zo serveru na endpoint v poslednom tagu).

Spomínaný modul už je využívaný spoločnosťou HoneyWell pre jednu prototypovú aplikáciu využívajúcu BLE technológiu implementujúcu technológie uART a Beacon.

LITERATÚRA

- [1] Roy Want *BLE Eddystone Beacon Service*, 2015. Opis Eddystone rámca.
- [2] *Understanding the different types of BLE Beacons*, 2015. Rozdiely medzi beaconmi.
- [3] Belal Khan *Firebase Cloud Messaging for Android using PHP and MySQL*, 2016. Firebase Cloud Messaging for Android using PHP and MySQL.
- [4] hotelie *How Starwood Hotels Makes It Personal and Makes You Come Back For More*, 2017. Využitie beaconov.
- [5] Apple INC. *Apple About iBeacon on your iPhone, iPad, and iPod touch*, 2017. iBeacon podpora v iOS.
- [6] CSc. Doc. Ing. Karel Burda. *Extrakce řečového signálu z hluku pozadí ve spektrální oblasti*. PhD thesis, Vysoké učení technické v Brně. Fakulta elektrotechniky a komunikačních technologií, 2016.
- [7] embt. *Úvod do světa IoT a Beacons*, 2017. Využitie beaconov.
- [8] Google. *Firebase Introducing Firebase*, 2016. Introducing Firebase.
- [9] Google INC. *Google Beacons*, 2017. Stručne o beacnoch.
- [10] Google INC. *Google INC Eddystone format*, 2017. Eddystone EID.
- [11] Google INC. *Kontakt.io Eddystone Ephemeral Identifier*, 2017. Eddystone EID.
- [12] iBeacon. *iBeacon What is iBeacon? A Guide to Beacons*, 2015. iBeacon komunikácia.
- [13] JSON.org. *JSON Introduction JSON*, 2017. Introduction JSON.
- [14] Kontakt.io. *Kontakt.io What is a Beacon?*, 2017. Technické možnosti beacnov od Kontakt.io.
- [15] MySqL.org. *MySQL What is MySQL?*, 2018. What is MySQL.
- [16] php.net. *PHP What is PHP?*, 2017. What is PHP.
- [17] Realm.io. *Realm Realm Docs*, 2018. Realm Docs.
- [18] Tan Jin Soon. Qr code. *Synthesis Journal*, 2008:59–78, 2008.

ZOZNAM SYMBOLOV, VELIČÍN A SKRATIEK

ms	milisekundy
QR	Quick Response
RFID	Radio-Frequency Identification
UHF	Ultra High Frequency
HF	High Frequency
LF	Low Frequency
NFC	Near Field Communication
BLE	Bluetooth Low Energy
UUID	universal Unique Identifier
API	Application programming interface
SDK	Software development kit
AVD	Android Virtual Device
ADB	Android Debug Bridge
PHP	Hypertext Preprocessor
UI	User interface
JSON	JavaScript Object Notation
URL	Uniform Resource Locator
SHA	Secure Hash Algorithm
MAC	Media Access Control address
dB	Decibel

ZOZNAM PRÍLOH

A Obsah priloženého CD

53

A OBSAH PŘILOŽENÉHO CD

Nasledující zoznam obsahuje zoznam súborov na priloženom disku spolu s ich stručným popisom.

/	koreňový adresár priloženého DVD
186120.pdf	bakalárska práca
BakalarkaMeranie.zip	..	projekt Android aplikácie na meranie prijatého výkonu
BakalarkaMarketing.zip	projekt Android aplikácie cieleného marketingu
debug-apk.apk	inštalčka marketingovej aplikácie
serveroveSkripty	serverová časť
push	skripty potrebné na doručenie push notifikácie
Config.php	definovanie konštánt
DbConnect.php	prihlásenie k databázi pre Firebase
DbOperation.php	funkcie yabypečujúce cielenie notifikácie pre všetkých/užívateľ
GetRegisteredDevices.php	skript ktorý vráti všetkých registrovaných užívateľov
Push.php	finálne odoslanie notifikácie
RegisterDevice.php	priradenie tokenu k registrovanému užívateľovi
sendMultiplePush.php	odoslanie notifikácií všetkým registrovaným užívateľom
sendinglePush.php	odoslanie notifikácií jednému užívateľovi
dbBAK.php	prihlásenie k databázi
deal24.php	endpoint indikujúci trávenie času pri príslušnom beacone
deal42.php	endpoint indikujúci trávenie času pri príslušnom beacone
entry.php	endpoint indikujúci vstup do obchodu
getTags.php	endpoint vracajúci všetky uložené tagy
checkout.php	endpoint indikujúci platenie na pokladni
inside.php	endpoint indikujúci reálny vstup do predajne
login.php	overenie pri prihlásení
out.php	kontroly pri odchode z predajne
outDeal.php	odoslanie notifikácie pri odchode - vždy
register.php	registrácia nového užívateľa
whitchway.php	zistenie či užívateľ ide z alebo do predajne
readme.docx	postup konfigurácie projektu